

## KOREAN PATENT ABSTRACTS

(11)Publication number: **10-2005-0023242 A**  
(43)Date of publication of application: **09.03.2005**

(51)Int. Cl.

**G06F17/30; G06F17/60**

(21)Application number: **10-2004-7015700**

(22)Date of filing: **01.10.2004**

(71)Applicant: **OVERTURE SERVICES, INC.**

(72)Inventor: **ROY SCOTT**

**SINGH NARINDER P**

---

**(54) DISPLAYING PAID SEARCH LISTINGS IN PROPORTION TO  
ADVERTISER SPENDING**

(57) Abstract:

In a pay for placement database search system, in which advertisers pay to include their search listings in a database to be provided with search results in response to queries from searchers decides how much money he wants to spend on a search term. The search provider displays the advertisers' listings in proportion to the amount of money the respective advertisers spend. This permits the advertisers to subscribe to the database search system, deciding how much to pay for a subscription for a predetermined time period. The search provider can recommend an optimal spend amount for the advertisers.

**(19) 대한민국특허청(KR)**  
**(12) 공개특허공보(A)**

(51). Int. Cl.<sup>7</sup>  
G06F 17/30  
G06F 17/60

(11) 공개번호 10-2005-0023242  
(43) 공개일자 2005년03월09일

(21) 출원번호 10-2004-7015700  
(22) 출원일자 2004년10월01일  
번역문 제출일자 2004년10월01일  
(86) 국제출원번호 PCT/US2003/009528  
국제출원출원일자 2003년03월31일

(87) 국제공개번호 WO 2003/085561  
국제공개일자 2003년10월16일

(30) 우선권주장 60/369,460 2002년04월01일 미국(US)

(71) 출원인 오버추어 서비스, 인크.  
미국 캘리포니아 (우편번호: 91103) 파사데나 노쓰 파사데나 애비뉴 74 써드 플로어

(72) 발명자 로이. 스콧  
미국 94306 캘리포니아주 팔로 알토 넘버 105 아라스트라데로 로드  
724

싱, 나린더 피.  
미국 89451 네바다주 인클린 빌리지 넘버 11 노쓰우드 블러바드  
928

(74) 대리인 주성민  
백만기  
이종희

심사청구 : 있음

**(54) 광고주 지출에 비례한 유료 검색 리스트의 표시**

명세서

배경 기술

<관련 출원>

본 특허 출원은 2002년 4월 1일 출원된 미국 가특허 출원 제60/369,460호의 출원일의 이익을 청구하며, 그 전체가 본 명세서에 참조로 포함된다.

PFP(pay-for-placement) 검색 엔진은 옐로우 페이지의 온라인 버전과 같이 작동한다: 사용자는 검색을 수행하고, 시스템은 사용자의 질의에 매칭되는 유료 광고주 리스트를 표시한다. 스크린의 실제 면적이 제한되므로, 사용자는 통상 그의 질의에 매칭하는 리스트들 중 일부만을 보게 된다. 그래서, 실행 가능한 시스템을 만들기 위해, 검색 엔진 제공자들은 각 리스트를 얼마나 자주 표시할지, 얼마나 청구할지, 및 리스트가 나타나는 순서를 결정할 필요가 있다.

본 기술의 현재 수준은 www.overture.com에서 얻을 수 있는 오버추어의 검색 엔진이다. 오버추어는 순위 입찰(bid-for-rank)로서 알려진 스킁을 이용하는데, 이는 클릭에 의해 광고주에게 청구를 하고 각 광고주가 각 클릭에 대해 지불하려고 하는 액수에 기초하여 리스트를 순서화한다. 광고주는 그가 원하는 무엇이든지를 클릭당 최소 5센트의 입찰가로 입찰할 수 있다. 많은 웹 사이트들은 오버추어 검색 결과를 표시하고, 각각은 리스트의 최상위에서 시작하는 상이한 개수의 검색 결과들을 보여주므로, 최상위 근처에 있으려는 강한 동기가 존재한다. 최고 입찰가의 광고주는 항상 나타나고, 입찰가가 감소함에 따라 광고주는 점점 더 적게 나타난다. 이론적으로는, 최고의 서비스 품질을 제공하는 광고주가 최고로 입찰하고 리스트의 최상위에 나타난다. 실제로는, 시스템은 이렇게 잘 작동하지는 않는다. 이러한 스킁은 이론적으로는 간단하지만, 사용자 및 광고주 모두를 좌절하게 하는 다수의 문제들이 있다.

사용자의 관점에서, 문제는 순위 입찰은, 관련이 없고 원하지 않는 검색 리스트를 최상위 근처로 유도할 수 있다는 것이다. 광고주는 그 리스트가 나타나는 순서에 대한 절대적인 제어권을 가지고 있고, 영리한 광고주들은 이러한 자유를 이용하여 사용자의 비용으로 공짜의 노출을 얻을 수 있다. 예를 들면, 애완용 흰족제비를 파는 광고주가 있다고 상상해 보자. 광고주는 "흰족제비" 및 "애완용 흰족제비"와 같은 매우 특정된 검색어 하에서 나타나고, 또한 "애완동물"과 같은 더 일반적인 용어 하에서도 나타난다. 사용자가 "애완용 흰족제비"를 검색하는 경우, 사용자는 흰족제비 광고주가 팔고 있는 것을 찾고 있는 것이 거의 확실하므로, 흰족제비 광고주가 리스트의 최상위에 나타나는 것이 이치에 맞는다. 그러나, "애완동물"과 같은 더 일반적인 용어에 대해, 흰족제비 광고주는 최상위 근처에 나타나지 않는다. "애완동물"을 타이핑하는 사용자는 흰족제비보다는 개나 고양이를 찾을 가능성이 훨씬 크다.

불행하게도, 흰족제비 광고주가 영리하다면, "뛰어난 애완용 흰족제비!, 엄청 쌍니다!"라고 읽혀지는 리스트를 생성할 것이다. 그는 사용자가 이것을 클릭할 때 이것이 흰족제비에 관한 것이라고 보는 것을 알고 있으므로, 이러한 리스트에 대해서는 매우 높은 가격 - 이것을 리스트의 최상위로 가져가는 가격에 입찰할 수 있다. 리스트 내에는 광고주가 판매하고자 매우 원하는 충분한 정보가 있다. 순위 입찰 시스템에서, 광고주는 리스트의 최상위에 있는 것에 대해 지불하지 않고, 단지 사용자가 그 리스트에 실제로 클릭하는 경우에만 지불한다. 이러한 리스트는 매우 구체적어서 사용자는 흰족제비에 관심이 있는 경우에만 클릭할 것이고, 따라서, 리스트가 "애완동물"과 같은 일반적인 용어 하에 나타나는 경우라도 광고주에게는 거의 위협이 없다. 결론적으로, "애완동물"에 관한 제1 검색 결과는 소수의 사용자들에게만 관련된 리스트라는 것이다. 나머지 모든 사람에게는, 이 리스트는 무용지물이고, 전체 검색 경험에 빈약하다. 이러한 흰족제비 현상은 PFP 검색 엔진에서는 흔한 일이다. 엘로우페이지는 이러한 문제점이 없는 데, 이는 광고주가 이들이 페이지를 차지하는 양에 기초하여 지불해야 하기 때문이다.

광고주의 관점에서는, 순위 입찰의 문제점은, 복잡하고 그가 얻는 것을 알기가 어렵다는 것이다. 광고주가 특정 순위로 입찰한 경우, 미리 그가 얼마나 많은 클릭을 수신할지, 또는 그는 얼마나 많은 돈을 지출할지, 또는 그가 일부 다른 순위에서 보다 높은 이익을 가질지 여부를 계산할 어떠한 방법도 가지고 있지 않는다. 또한, 다른 광고주가 더 높은 입찰 가로 그 뒤로 들어올 수 있으므로, 심지어는 그가 원하는 순위를 얻는 것이 확실하지 않을 수 있다. 광고주가 한정된 예산을 가지고 있다면, 최대 가능한 클릭 개수를 얻기 충분한 입찰 가를 유지하면서도, 예산을 초과하지 않도록 그 지출을 계속해서 모니터링해야 한다. 통상, 각각이 그 자신의 입찰 가를 가지고 있는 50 또는 100개의 다른 검색 용어에 대해 모든 이러한 불확실성을 다루어야 한다.

예로서, 광고주가 검색어 "신선한 생선"에 대해 두 번째 위치에 있도록 현재 \$1.00을 입찰하고 있다고 가정하자. 그는 현재 그 위치에 유지하거나, 입찰 가를 \$1.20으로 올려 순위 1로 올라가거나, 입찰 가를 \$0.80으로 내려 순위 3으로 내려갈 수 있다. 이러한 결정을 내리기 위해서는, 3개의 순위들의 각각에서 그의 것이 얼마나 많이 클릭되는지를 알 필요가 있다. 그가 그러한 정보를 얻는 것은 거의 불가능하다. 본 기술분야의 현재 상태를 보면, 사실상 검색 엔진 제공자들은 그가 순위 3보다 순위 1에서 더 많은 클릭을 얻을 것이라는 것을 확실하게 말할 수도 없다. 광고주가 순위 1로 입찰 가를 올리는 것이 타당하다고 결정하더라도, 다른 광고주들 중 하나가 수시간 후에 그보다 입찰 가를 높게 하지 않을 것이라는 보장도 없다. 또는, 동일하게 광고주는 순위 1로 입찰 가를 올리고나서, 수일 후에 그 아래에 있는 광고주들이 없어졌으므로 과도하게 지불하고 있다는 것을 발견할 가능성도 있다. 광고주는 과도하게 지불하지 않고 그가 원하는 것을 얻기 위해 그의 입찰 가와 위치를 연속적으로 모니터링해야 한다. 본 기술분야의 현재의 상태는 전자 입찰 에이전트를 이용하여 입찰을 추적하는 것이다. 그 예들은 [www.gotoast.com](http://www.gotoast.com), [www.did-it.com](http://www.did-it.com), 및 [www.pay-per-click-bid-managers.com](http://www.pay-per-click-bid-managers.com)이다. 그러나, 이들은 단지 주기적으로 실행되므로 이들이 얼마나 잘 수행하는지에 대해 제한적이고, 검색 엔진으로부터 그들이 필요로 하는 정보, 예컨대 광고주가 상이한 순위에서 받기를 원하는 클릭의 회수와 같은 정보를 얻을 수 없다.

광고주가 고정된 예산을 가지고 있다면, 그는 연속적으로 그 지출을 모니터링하여, 그가 그 예산을 충족하도록 해야 한다. 예를 들면, 광고주가 다음 달에 지출할 \$1,000을 가지고 있다고 가정하자. 그가 그 입찰 가를 \$1.00으로 설정하고 50명의 사용자가 첫날에 그 리스트를 클릭한다면, 지출 비율이 너무 높으므로 입찰 가를 낮추어야 한다. \$50/일이라면, 그는 월말이 전에 그 전체 예산을 소모할 것이다. 반대로, 단지 10명의 사용자들만이 그 첫날에 리스트를 클릭한다면, 그 지출 비율이 너무 낮으므로, 입찰 가를 높여야 한다. 입찰과 예산간의 상호작용은 복잡하고, 일정한 조정없이 바로 얻기가 어렵다. 이는 또한 최상의 가장 관련있는 광고주를 보는 것 대신에 사용자는 단지 현재 그 예산 하에 있는 광고주들을 보고 있으므로, 나쁜 검색 결과를 유도할 수 있다.

이러한 모든 문제들은 광고주가 복수의 검색어에 입찰하는 경우에 훨씬 더 복잡하게 된다. 각 용어는 상이한 개수의 검색 및 클릭을 수신하고, 상이한 입찰 가를 요구한다. 광고주는 그 전체 이익을 최적화하는 방식으로 그 돈을 이들 사이에서 어느 정도 할당해야 한다. 광고주들이 이것을 수행하는데 도움을 주는 바람직한 도구는 현재 존재하지 않으며, 최상의 입찰 에이전트도 고정된 예산에 매칭되도록 복수의 용어들 사이에서 입찰 가를 높이거나 낮추려고 시도하지 않는다.

많은 광고주들은 이러한 시스템에 대한 대안을 선호한다. 정교한 입찰 구조는 너무 많은 극소 관리를 요구하고, 이는 광고주가 실제 염려하는 단지 2개의 문제, 즉 그가 지불해야 할 비용 및 이에 대한 보답으로 얻는 것을 흐리게 한다. 광고주는 검색 엔진 제공자가 \$1,000에 대해 다음 달에는 1,000 클릭을 구입할 수 있고, 또는 2배만큼 비용을 지출하면 2배의 클릭 회수를 얻을 수 있으며, 절반의 비용을 지출하면 절반의 클릭 회수를 얻을 수 있다고 말하기를 원한다. 이러한 모델은 엘로우페이지 및 인터넷 배너 광고와 같은 다른 종류의 광고와 훨씬 더 조화된다. 정보가 추출되어 비용과 클릭이 된다면, 입찰 및 순위라는 전체 개념이 불필요하다는 것은 명백하다. 광고주는 실제로 검색 결과 리스트에서 어떤 순위에 나타날지에 대해 관심을 가지지 않는다. 그는 그가 투자한 금액에 대해 얼마나 많은 클릭을 얻을지, 및 이를 클릭이 얼마나 자주 판매로 연결될지에 대해 관심을 가진다. 이 상적으로는, 그는 이러한 정보, 즉 그가 지불하는 액수, 및 그 보답으로 그가 얻는 클릭의 회수만에 기초하여 이러한 구매 결정을 하고, 그 리스트가 나타나는 위치 및 때에 관한 다른 모든 세부 사항은 검색 엔진 제공자에게 맡길 수 있다. 이러한 아이디어에 기초한 시스템은 사용자의 질의에 응답하여 어느 리스트를 보여주어야 할지를 결정하는데 검색 엔진 제공자에게 완전한 자유를 제공할 수 있으므로, 광고주에게 훨씬 더 단순할 뿐만 아니라 흰족제비 문제도 해결한다.

본 실시예들은 순위 입찰 스킴의 최상의 특징들을 유지하면서 입찰 및 순위를 제거한 것으로서, 경매를 이용하여 자동으로 가격을 설정하고 광고주가 검색어를 이들이 나타내야 할 곳을 골라낼 수 있도록 한다. 뿐만 아니라, 이들 실시예들은 광고주가 나타나는 곳에서 검색어들에 대한 각 광고주의 지출을 자동으로 최적화하므로, 입찰 에이전트를 필요로 하지 않는다.

이들 실시예들의 아이디어는 간단하다. 각 광고주는 검색어에 얼마나 많은 돈을 지출하기 원하는지를 결정하고, 검색 제공자는 각 광고주가 지출하는 돈의 액수에 비례하여 광고주의 리스트를 표시한다. M개의 리스트를 표시하는 검색 결과 페이지 상의 공간에 대해 경쟁하는 N명의 광고주가 있다고 가정하자. T개의 검색이 있다면, 검색 제공자는 광고주들에게 분배하는 TM개의 전체 임프레션(impression)을 가지고 있다. 임프레션은 검색자에게 제공되는 검색 결과들 중 하나의 검색 리스트의 표시이다. 광고주  $i$ 가 기꺼이 지출하는 돈의 액수가  $a_i$ 라면, 그가 받는 임프레션의 개수는 수학식 1과 같다.

그가 예상할 수 있는 클릭의 회수는 수학식 2와 같다.

형  $r_i$ 는 광고주의 클릭-스루 레이트(click-through rate)로서, 검색시 나타난 경우에 사용자가 그 리스트를 클릭하는 확률을 나타낸다.

정확한 임프레션 개수를 모든 사람에게 분배하기 위해, 제공자는 각 광고주가 얼마나 많은 임프레션을 수신하는지를 나타내는 실행 전체  $s_i$ 를 유지한다. 사용자가 검색을 수행하는 경우, 제공자는 이 회수와 광고주가 수신되어야 하

는 임프레션 개수간의 차이를 계산한다. 이러한 차이는  $DISCREPANCY_i = s_i - t_i \frac{S}{T}$  이고, S는 발생하는 전체 검색 회수이다. 제공자는 광고주들을 차이에 의해 분류하고, 최상위 M을 리턴한다. 리턴된 리스트의 순서는 임의적이거나, 클릭-스루 레이트  $r_i$ 에 의해 분류되거나, 리스트의 파악된 품질에 의해 분류되거나, 일부 다른 기준에 의해 분류될 수 있다. 이러한 알고리즘은 임프레션을 시간에 걸쳐 고르게 분배하면서 T개의 전체 검색 이후에 정확한 임프레션 개수를 수신하는 것을 보장한다.

광고주가 구매할 수 있는 임프레션의 개수에도 한계가 있다. 하나의 실시예에서, 어떠한 광고주도 임의의 검색 결과에서 한번 이상 나타날 수 없으므로, T 이상의 전체 임프레션을 가질 수 없다. 모든 광고주가 지출하는 총 금액을 T라 하면, 광고주에게 제한되는 금액은 수학식 3과 같다.

우리는 이러한 한계를 트래픽 한계로서 지칭한다. 광고주가 트래픽 한계 이상이라면, 검색 엔진 제공자는 여분의 임프레션을 공백으로 남겨둔다.  $N \leq M$ 이라면, 트래픽 한계에 있는 적어도 하나의 광고주가 항상 있고, 시장이 불평화된다. 이 경우에,  $M < N$ 이고 광고주들이 경쟁할 동기를 가지도록, 검색 엔진 제공자는 리턴 할 리스트의 개수를 감소시킬 수 있다. 다르게는, 제공자는 항상 리스트의 개수를 광고주 개수의 일부로, 예를 들면  $M=0.5N$ 이 되도록 설정할 수 있다.

양호한 실시예에서, 광고주는 임프레션이 아니라, 클릭 또는 클릭스루에 대해 지불하고, 가격은 판매 포인트에서 확정된다. 클릭 또는 클릭스루는 검색자가 광고주의 검색 리스트를 보고 그 연관된 하이퍼링크를 클릭하거나, 다르게는 검색 리스트를 보기 위해 선택하는 액션이다. 검색자의 웹 브라우저는 검색 리스트와 연관된 유니폼 리소스로 케이터(URL)로 재지향된다. 광고주가 클릭한 것을 구매하고자 원하는 경우, 검색 엔진 제공자는 사용자가 광고주의 리스트에 클릭하는 예상 회수 및 판매할 것으로 예상되는 전체 광고에 기초한 클릭당 비용(cost-per-click)으로 산

$CPC_i = \frac{\hat{A}}{TM_i^{\hat{A}}}$  출한다. 제공자가 가격을 매기는 클릭당 비용은  $\hat{A}$ 는 T 개의 검색에 대해 제공자가 판매하려고 하는 광고의 전체 액수 A에 대한 검색 엔진 제공자의 추정이고,  $t_i$ 는 광고주의 클릭-스루 레이트의 추정이다. 광고주의 클릭당 비용은 그 예상된 클릭-스루 레이트가 내려감에 따라 올라간다. 모든 사람이 동일한 클릭-스루 레이트를 가지고 있다면, 모든 사람은 동일한 CPC를 지불한다. 검색 엔진 제공자는 그가 전달할 것으로

$ESTIMATED\_CLICKS_i = TM_i \frac{a_i}{\hat{A}}$ 로하여 가격을 매긴다. 판매를 완료하기 위해, 광고주는 표시된 클릭당 비용에서 지출하기를 원하는 액수  $a_i$ 를 결정한다. 광고주가 지출 한계에 도달하거나 사용자들이 T 전체 검색들을 종료할 때까지, 제공자는 이 가격으로 클릭을 전달한다.

검색 엔진 제공자는 추정  $\hat{A}$ 를 조정함으로써 가격을 제어한다. 그는 추정이 가능한 한 정확한 경우에, 즉  $\hat{A}=A$ 인 경우에 가장 많은 소득을 올린다. 추정  $\hat{A}$ 가 너무 작으면, 클릭 추정이 너무 크고, 광고주는 개별적인 한계에 도달하지 않은 상태로  $\hat{A}$ 를 지출한다. 이 경우에, 제공자는  $\hat{A}$ 를 증가시킴으로써 가격을 올려야 한다. 광고주는 더 적게 지출하려는 경향이 있지만, 제공자가 만드는 전체 액수  $\hat{A}$ 는 올라간다. 추정  $\hat{A}$ 가 너무 크면, 클릭 추정이 너무 작고 광고주는 T 검색의 종료 이전에 전체 한계 A에 도달한다. 이 경우에, 제공자는  $\hat{A}$ 를 낮춤으로써 가격을 줄여야 한다. 광고주는 더 많이 지출하려는 경향이 있고, 제공자가 만드는 전체 액수 A도 올라간다. 추정  $\hat{A}$ 가 정확하다면, 각 광고주는 어떠한 미판매된 클릭없이 개별적인 지출 한계  $a_i$ 를 지불한다. 마찬가지로, 검색 엔진 제공자도 모든 추정된 클릭-스루 레이트가 가능한 한 정확한 경우에 가장 성공한다.

전체 지출 및 클릭-스루 레이트의 양호한 추정을 하는 하나의 방법은 역사적 데이터를 이용하는 것이다. 예를 들어,

$$r_i = \frac{u_i + k \text{CTR}}{s_i + k}$$

전체 시장에 대한 역사적 클릭-스루 레이트가 CTR이라고 가정하자. 그러면, 제공자는 공식하여 개별적인 리스트의 클릭스루 레이트를 추정할 수 있다. 항  $s_i$ 는 리스트가 수신한 클릭의 회수이고,  $s_i$ 는 임프레션의 개수이다. 새로운 리스트에 대해, 이들 개수들은 제로이고, 추정은 전체 클릭-스루 레이트 CTR로 초기화된다. 다수의 클릭 및 임프레션을 수신한 리스트에 대해, 추정은 관찰된 빈도에 접근한다. 상수  $k$ 는 추정이 이들 2개의 값 사이에서 얼마나 신속하게 변경하는지를 결정하는 스케일링 인자로서 작용한다. 광고주가 그 리스트에 대해 더 많은 클릭을 구매하려고 결정한 경우, 제공자는 지금까지 최고의 추정  $r_i$ 에 기초하여 클릭당 비용을 전적한다. 제공자는 다른 방식, 예를 들면 새로운 리스트와 역사적 데이터인 다른 리스트와 비교함으로써 클릭-스루 레이트를 추정할 수 있다.

제공자가 광고주를 등록하는 프로세스를 실행하는 다수의 다른 방법들이 있다. 예를 들어, 제공자는 주별 또는 월별 경매를 실행하여 액수  $a_i$ 를 확정하거나, 광고주가 이를 연속적으로 조정할 수 있게 한다. 이러한 공개는 임의의 특정 스킴으로 제한되지 않는다.

검색 결과를 표시하는 모든 웹 사이트는 다르다. 본 기술분야의 현재 상태에서, 리스트의 개수는 1 내지 20 사이에서 가변하고, 일부 웹 사이트는 완전한 리스트를 표시하지만, 다른 웹 사이트들은 타이틀만을 표시한다. 이를 차이점으로 인해, 바람직한 실시에는 검색어 및 웹 사이트의 모든 조합을 분리된 시장으로서 다룬다. 대안은 어느 곳에서든지 검색어가 수신하는 모든 임프레션을 단일 시장으로서 다룬 것이다. 이 경우에, 트래픽 한계 공식은 수학식 3과 다르지만, 나머지 수학식 및 알고리즘은 동일하게 유지된다. 당업자라면, 이러한 대안적인 공식에 대해 적절한 새로운 트래픽 제한 공식을 도출하는데 아무런 문제가 없다.

광고주는 통상 다수의 다른 검색어에 대해 돈을 지출한다. 본 발명의 하나의 장점은, 제공자는 광고주가 지출하는 최적의 액수를 추천할 수 있고 다른 시장들 사이에서 그 돈을 자동으로 할당할 수 있다는 점이다. 광고주가 제공자의 추천을 따르는 경우, 그는 예상되는 이익을 최대화하도록 보장된다. 본 섹션의 나머지는 단일 시장에서 단일 광고주의 경우로부터 시작하여 다수의 시장에서 다수의 광고주의 모든 문제에까지 적용하는 필요한 공식 및 알고리즘을 기술한다.

광고주의 지출을 최적화하기 위해, 제공자는 3가지, 즉 광고주의 전체 예산  $b_i$ , 광고주의 클릭당 이익(profit-per-click)  $p_i$ , 및 광고주들이 다른 곳에서 돈을 지출하는 경우에 모든 광고주에게 가용한 외부 리턴 레이트  $R$ 을 알아야 한다. 단일 시장에서 유료 리스트를 구입하는 것으로부터 광고주의 예상 이익은 수학식 4와 같다.

첫번째 항은 광고주의 유료 리스트로부터의 광고주의 이익이고, 두번째 항은 다른 곳에 돈을 투자한 것으로부터 그의 이익이다. 외부 리턴 레이트  $R$ 이 높으면, 광고주는 클릭에 대해 그의 예산 중 더 적은 부분을 지불한다. 그의 이익을 최대화하는  $a_i$ 의 값은 수학식 5이다.

양  $d_i$ 는 시장에서 다른 광고주에 의해 지출되는 돈의 전체 액수이고,  $v_i = r_i p_i$ 는 광고주의 시장 값이다. 광고주가 지출하는 액수는 그 클릭-스루 레이트 또는 그 클릭당 이익이 감소됨에 따라 내려간다.  $a_i$ 가 제로보다 작거나, 제공자의 최소 지출액보다 작거나, 트래픽 한계보다 크거나, 광고주의 예산보다 크면, 이는 적절한 값으로 제한된다.

복수의 시장이 있는 경우, 최적의 솔루션은 이들을 독립적으로 다룬다는 것이다. 광고주의 최적의 지출은 각 시장에 대한 수학식 5의 하나의 예에 의해 주어진다. 양  $T$ ,  $v_i$ , 및  $d_i$ 는 각 시장에서 모두 상이할 수 있다. 시장들간의 유일한

$$\sum a_i \leq b_i$$

상호작용은 광고주의 예산 한계  $'$ 이다. 이 합은 모든 시장에 대해 취해진다. 이러한 한계를 강제하기 위해, 광고주의 시장 값은 예산 스케일링 인자  $v_{ij} = \lambda_i r_{ij} p_{ij}$ 를 포함한다. 유의할 점은, 모든 시장에서 광고주의 값은 일정하게 스케일링하는 하나의  $\lambda_i$ 가 있다는 점이다. 광고주가 예산 제한되지 않는다면,  $\lambda_i = 1$ 이다. 광고주가 예산 제한된다면,  $\lambda_i$ 는 그 예산을 정확하게 지출하는 값으로 설정된다. 임의의 양호한 숫자 방정식 해결자라면 이 값을 구할 수 있다. 예들은 [www.ece.nwe.edu/OTC](http://www.ece.nwe.edu/OTC) 및 *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Dennis and Schnabel, ISBN 0898713641에서, 비선형 방정식 해결자에 대한 온라인 지침을 포함한다.

하나의 시장에서 복수의 광고주가 있다면, 문제는 더 복잡하다. 임의의 광고주가 그가 기꺼이 지불하고자 하는 액수를 변경하면, 다른 광고주들에 대한 최적의 솔루션은 변경된다. 하나의 가능한 알고리즘은 각 광고주에 대해 반복적으로 구하고, 반복이 수렴하기를 바라는 것이다. 이들이 수렴한다면, 최종 해답은 다른 광고주가 기꺼이 지불하고자 하는 액수를 변경하지 않는 경우에는 어떠한 광고주도 그 이익을 증가시킬 수 없는 확정된 포인트이다. 수학식 4로부터  $f_i$ 에 대한 정의를 이용하면, 이러한 확정된 포인트는 수학식 6의 방정식 시스템을 만족한다.

하나의 광고주가 그가 지출하는 액수를 변경하면 다른 광고주들에 대한 이익이 올라가므로, 확정된 포인트는 임의의 광고주에 대한 포괄적인 최대치가 아니다. 여전히, 이것은 그가 제어할 수 있는 하나의 양 - 자신의 지출 -에 대해 각 광고주의 이익을 최대화하므로, 이는 문제에 대한 정확한 솔루션이다.

수학식 6의 시스템을 닫힌 형태로 해결할 수 있다. 각 광고주에 대한 솔루션은 수학식 7과 같다.

모든 광고주가 함께 지출하는 전체 달러 액수 A는 수학식 8과 같다.

양  $v_L$ 은 수학식 7이 음이 되기 전에 광고주가 가질 수 있는 최저 시장 값이고, 광고주는 어떠한 돈도 지출하지 않는다. 그 값은 모든 광고주 값의 조화 평균에 관련된다.

여기에서, 조화 평균 값  $\bar{v}$ 는 수학식 10과 같다.

다른 관심있는 양은  $v_U$ 로서, 광고주가 트래픽 한계가 되기 이전에 가질 수 있는 최대 값이다. 그 값은 수학식 11과 같다.

수학식 7-11은 이를 실시예들의 기본적인 공식이다.

수학식 7을 이용하여 최적 지출액을 계산할 때, 일부 광고주는 제로 또는 트래픽 한계보다 적은 최적 액수를 가지는 것이 가능하다. 이를 솔루션들은 실제 세계에서는 불가능하다. 유효한 솔루션을 구하는 알고리즘은 경계 외부에 있는 광고주들 중 하나를 집어내고, 이를 적절한 한계로 제한하며, 이를 문제로부터 제거하고, 다른 모든 다른 사람에 대한 솔루션을 재계산하는 것이다. 각 반복에서 재계산된 값은  $v_L$ 에 대해 약간 다른 공식을 이용한다.

이러한 수학식에서, m은 트래픽 한계에서 광고주들에 의해 차지되지 않는 검색 결과 페이지 상의 자유 공간의 개수이고, n은 제로 한계 또는 트래픽 한계 중 어느 하나가 아닌 자유로운 광고주의 개수이다. 조화 평균  $\bar{v}$ 는 자유로운 광고주들에 대해 계산된다. 모든 광고주들이 자유롭다면,  $m=M$ 이고  $n=N$ 이며, 그 솔루션은 앞서 수학식 9의 공식과 동일하다.

제거할 광고주를 집어내기 위한 알고리즘은 음이거나 트래픽 한계를 초과하는 전체 부호가 있는 파도 지출을 본다.

$E < 0$ 이면, 제로 한계 이하인 광고주들에 대해 다수의 파도한 지출이 있고, 알고리즘은 최저 시장 값을 가지는 광고주를 제거한다.  $E \geq 0$ 이면, 트래픽 한계를 넘는 광고주들에 대해 다수의 파도한 지출이 있고, 알고리즘은 최대 시장 값을 가지는 광고주를 트래픽 한계로 제한한다. 경계 경우  $E=0$ 은 시장이 검색 결과 페이지 상의 공간보다 더 적은 광고주들로 붕괴되지 않게 보장하도록 트래픽 한계를 부가하도록 결정된다.

복수의 시장 및 복수의 광고주가 있는 경우, 최적의 솔루션은 이를 독립적으로 다루는 것이다. 단일 광고주에서와 같이, 각 광고주의 시장 값은 모든 시장에 걸친 그의 지출을 제한하는 예산 스케일 인자를 포함한다. 광고주가 예산 제한되지 않는다면, 그의 예산 스케일 인자  $\lambda_i = 1$ 이다. 광고주가 예산 제한된다면,  $\lambda_i < 1$ 이다. 첫번째 k가 예산에 제한되는 것들이 되도록 광고주를 다시 번호를 매기려면, 예산 제한을 만족하는 알고리즘은 비선형 방정식의 동시 시스템을 푸는 것이다.

자유 변수는 그 예산 한계에서 광고주들에 대한  $\lambda_i$ 이다.  $a_{ij}$ 의 값들은 수학식 7 및 12에 의해 주어지고, 한계 0 및  $A_j/M_j$ 에 의해 제한된다. 임의의 뛰어난 방정식 해결자라면 이 시스템을 수치적으로 풀 수 있다.

몇 가지 실질적인 문제가 있다. 우선, 광고주의 예산 스케일 인자가 변경됨에 따라 광고주의 시장 값이 변경되므로, 그의 최적 지출은 수학식 14의 시스템을 해결할 때 제로 한계 또는 트래픽 한계 중 하나를 교차할 수 있다. 그러므로, 광고주가 처음에 일부 시장에서 제한되더라도, 수학식 7 및 12를 이용하여  $a_{ij}$ 를 항상 계산하는 것이 중요하다. 항상 수학식들을 이용함으로써, 광고주의 전체 최적 지출이 제로와 그 최대 값 사이에서 원활하게 가변된다.

두 번째로,  $\lambda_i$ 에 대해 풀 때 시장 제한이 변경될 수 있으므로, 알고리즘은 안정된 솔루션에 도달될 때까지 반복되어야 한다. 알고리즘은, 우선 각 시장에서 모든 제로 제한 및 트래픽 제한에 대해 해결하고, m 및 n 값을 구하며, 예산 제한을 해결하며, m 및 n 값을 확정되게 유지하는 것이다. 임의의 광고주가 예산 한계에 대해 해결하는 동안에 제로 한계 또는 트래픽 한계를 교차하는 경우, 프로세스를 반복한다.

세 번째로, 알고리즘은 예산 제한에 대해 해결할 때 수학식 14에 어느 광고주를 포함할지를 결정해야 한다. 알고리즘은 예산이 초과한 임의의 광고주 및  $\lambda_i < 1$ 인 임의의 광고주를 포함한다. 이러한 조합은 그 예산 한계에 있는 광고주의 워킹 세트를 제공한다. 이러한 워킹 세트는 잘못된 것일 수 있고, 일부 광고주들은  $\lambda_i = 1$ 인 경우라도 실제로는 예산 이하일 수 있다. 이 경우에, 수학식 14는 현재 워킹 세트로는 솔루션을 가지고 있지 않는다. 뛰어난 방정식 해결자라면 가능한 한 많은 예산 제한에 대해 해결할 수 있다. 다음 반복에서, 알고리즘은 워킹 세트에 속하지 않는 광고주들을 제거하고 수학식 14를 해결한다.

마지막으로, 광고주의 전체 최적 지출이 그 예산 스케일이 변경됨에 따라 변경되지 않는 영역들이 있다. 이러한 상황은 광고주가 모든 시장에서 제한될 때 발생한다. 도 21은 그 문제를 예시하고 있다. 본 도는 광고주의 전체 최적 경비가 그 예산 스케일의 함수로서 가변되는 방법을 플롯팅한다. 광고주의 최적 지출이 모든 시장에서 제로 한계 이하이므로, 그래프는 좌측에서 편평하고, 그 최적 지출이 모든 시장에서 트래픽 한계 이상이므로, 그래프는 우측에서 편평하다. 이를 편평한 영역들은 숫자 방정식 해결자에 대해 문제를 유발시킬 수 있다.

솔루션은 수학식 해결자에게  $\lambda_i$ 에 대한 경계 제한으로서 2개의 포인트  $\min \lambda_i$  및  $\max \lambda_i$ 를 제공하는 것이다. 이들 경계들은 제로 및 1의 통상적인 한계를 대체한다. 단일 시장 내에서,  $\min \lambda_i$ 는 광고주가 제로 한계에 도달하는 포인트이고,  $\max \lambda_i$ 는 광고주가 트래픽 한계에 도달하는 포인트이다. 이들 값들은 수학식 15와 같다.

$\beta$ 의 값은 다음과 같다.

$\beta = 0$  광고주가 제한된 경우

$\bar{v}/n$  광고주가 자유로운 경우

원하는 경계 제한은 모든 시장에 걸친 이들 값들의 최소 및 최대이다. 이러한 제한으로도, 수학식 해결자는 지출 그 래프의 중간에서 편평한 영역을 만날 수 있다. 이들 편평한 영역은 광고주가 일부 시장에서는 제로 한계 이하이고 나머지에서는 트래픽 한계 이상인 경우에 드물게 발생한다. 이 경우에, 알고리즘은 진행할 수 있도록 편평한 영역 외부에 있는  $\lambda_i$ 값으로 방정식 해결자를 재시작하는 것이다.

광고주는 자동적인 최적화를 무시하고, 그 지출을 자신이 조정할 수 있다. 모든 수학식들은  $v_L$ 에 대한 공식이 변경되는 것을 제외하고는 나머지 광고주들에 대해 계속해서 유효하다. C가 고정된 액수를 지출하는 광고주에 의해 제공되는 전체 금액이라면,  $v_L$ 은 이하와 같다.

양  $v_o$ 는 수학식 12로부터의 이전 제로 한계 값이다.

$$v_o = \bar{v} \left( 1 - \frac{m}{Mn} \right)$$

유의할 점은,  $C=0$ 인 경우에  $v_L$ 은  $v_o$ 로 감소된다는 점이다. 다른 관심있는 테스트는  $n=1$ 이고  $m=M$ 이다. 이 경우에,  $v_o=0$ 이고, 수학식 16은 수학식 5의 원래의 하나의 광고주 결과로 다시 유도된다.

하나의 실시예에서, 가입 방법은 PFP 데이터베이스 검색 시스템에 대해 구현된다. 이 방법은 광고주들에게 특정 비용으로 다수의 검색자 계약을 제공하는 검색 제공자를 포함한다. 검색자 계약은 임프레션, 클릭 또는 클릭스루, 포스트-클릭스루 액션 또는 일부 다른 티입의 계약일 수 있다. 이와 같이, 광고주는 이들이 기꺼이 지불하는 클릭의 개수 또는 액수, 및 따라서 가입자를 선택할 수 있다. 가입은 1개월과 같이 설정된 기간 동안이고, 임의의 상호 동의할 수 있는 방식으로 배열될 수 있다. 검색 제공자는 다른 광고주에 대해 다른 비율을 제공하거나, 광고주가 참여하는 검색 리스트 또는 시장의 개수에 따라 다른 비율을 제공할 수 있다.

이 방법은 가입하는 광고주들로 가입 어카운트를 개시하는 것을 더 포함한다. 가입 어카운트는 광고주들로부터 지불 신용되어 있고, 검색자 계약이 발생함에 따라 후속적으로 자동으로 인출된다.

PFP 데이터베이스 검색 시스템은 후속적으로 검색자들로부터 검색 요구를 수신한다. 이를 검색 요구에 응답하여, 데이터베이스 검색 시스템은 검색 결과를 제공한다. 일부 검색 결과들은 검색 질의가 검색 리스트와 매칭되는 경우에 가입 광고주의 리스트일 수 있다. 가입 광고주 검색 리스트가 검색 리스트의 페이지로 제공된다면, 가입 어카운트는 자동으로 조정된다. 이것은 어카운트에서 가입 광고주에 의해 지불되는 임프레션 또는 클릭스루의 개수를 세고 검색자에게 제공되는 각 검색 리스트에 대해 하나씩 인출함으로써 수행될 수 있다. 가입 관리의 임의의 다른 분류가 이용될 수 있다.

#### 도면의 간단한 설명

도 1-20은 각 광고주의 최적 지출을 구하기 위한 세부 알고리즘을 도시한 흐름도이다.

도 21은 광고주의 전체 최적 지출이 광고주의 함수로서 가변되는 것을 도시한 플롯이다.

도 22는 컴퓨터 데이터베이스 검색 시스템을 포함하는 네트워크의 하나의 실시예를 예시하는 블록도이다.

#### 실시 예

이제, 도면을 참조하면, 도 1-20은 각 광고주의 최적 지출을 구하기 위한 세부 알고리즘을 제공하는 플로우 다이어 그램을 형성한다. 알고리즘의 입력은 다음과 같다.

SEARCHES[M] : 시장 M에서의 검색의 회수

SPACES[M] : 검색 결과 페이지 상의 공간의 개수

ROM[M] : 외부 리턴 레이트

PROFIT\_PER\_CLICK[A,M] : 광고주의 클릭당 이익

CLICK\_RATE[A,M] : 광고주의 클릭-스루 레이트

BUDGET[A] : 광고주의 예산

알고리즘의 출력은 다음과 같다.

SPEND[A,M] : 광고주의 최적 지출액

CONSTRAINT[A,M] : 광고주의 제한 상태

LAMBDA[A] : 광고주의 예산 스케일 인자

상세한 설명 전체에 걸쳐, 변수 A는 항상 광고주를 지칭하고, 변수 M은 항상 시장을 지칭한다. BUDGET[A] 및 LAMBDA[A]를 제외한 모든 양이 현재 시장 M에 의존하므로, 그 가독성을 향상시키기 위해 도면에서는 통상 M을 생략한다. 마찬가지로, 도면은 예리 조건 또는 부동소수점 경계 조건, 또는 캐싱 또는 최적화를 위한 기회를 보여주지 않는다. 본 기술분야의 숙련자라면 의사코드를 해석하고 이를 구현하는 효율적인 컴퓨터 프로그램을 저작하는데 전혀 어려움이 없을 것이다.

여기에서 설명된 실시예들은 하나 이상의 처리 디바이스 및 연관된 데이터 저장 장비를 동작시키기 위한 컴퓨터 판독 가능한 프로그램 코드로서 구현될 수 있다. 하나의 특정 실시예에서, 개시된 방법 및 장치는 데이터베이스 관리 시스템 또는 검색 엔진을 제어하기 위한 C++ 프로그램 코드로서 구현될 수 있다. 다른 실시예에서, 방법 및 장치는 컴퓨터 판독 가능한 프로그램 코드를 저장하는 데이터 저장 매체, 여기에 설명된 기능을 수행하는 데이터 처리 장치 또는 임의의 다른 적합한 디바이스로서 구현될 수 있다.

도 1은 상위 레벨 방법의 하나의 실시예를 도시하고 있다. 본 실시예의 방법은 각 시장에서 제로 및 트래픽 제한에 대해 풀고 나서 예산 제한을 푸는 루프이다. 루프는 현재의 솔루션이 모든 제한을 만족한다고 할 때 IS\_SOLVED가 되는 경우 종료한다. 최종 단계에서, 알고리즘은 각 광고주의 최적 지출을 출력 변수 SPEND[A,M]에 기록한다.

이 상위 레벨 방법은 블록 100에서 시작하는 Solve라고 라벨링된 절차이다. 블록 102에서, 절차 INITIALIZE\_BUDGET\_SCALES가 호출된다. 이 절차는 광고주의 예산 스케일 인자, 람다를 초기화한다. 본 절차의 하나의 실시예는 도 9에 도시되어 있다. 루프는 블록 104에서 시작한다. 블록 106에서, 루프 변수가 초기화된다. 블록 108에서, 절차 SOLVE\_MARKET이 호출된다. 본 절차의 하나의 실시예는 도 2에 도시되어 있다. 이러한 절차는 시장에서 제로 및 트래픽 제한을 만족시킨다. 모든 시장 M이 처리될 때까지 블록 110에서 루프가 계속된다. 블록 112에서, 루프가 야기되어 절차 SOLVE\_BUDGET이 호출된다. 이러한 절차의 하나의 실시예는 도 10에 도시되어 있다. 이러한 절차는 각 광고주의 예산 제한을 만족시킨다. 블록 104, 106, 108, 110, 112, 및 114를 포함하는 루프는 절차 IS\_SOLVED가 참 값을 리턴할 때까지 처리를 계속한다. 이 절차의 하나의 실시예는 도 17에 도시되어 있다. 이러한 절차는 모든 제로, 트래픽 및 예산 제한이 해결되어 있는지를 결정한다. 그렇지 않다면, 제어는 블록 104로 되돌아간다. 그렇다면, 블록 116에서, 절차 COMPUTE\_SPENDING이 호출된다. 이러한 절차는 각 광고주의 최적 지출을 계산한다. 이러한 절차의 하나의 실시예는 도 18에 도시되어 있다. 그리고 나서, 절차 Solve는 블록 118에서 종료한다.

도 2-8은 제로 및 트래픽 제한을 푸는 단계들을 도시하고 있다. 출력은 시장에서 광고주가 제로 제한되거나, 트래픽 제한되거나 제한되지 않은지 여부를 나타내는 CONSTRAINT[A,M] 값이다. 상위 레벨 루프는 도 2에 도시되어 있다. 알고리즘은 처음에 현재 시장에서의 모든 제한들을 NONE으로 초기화하고, 그리고 나서 MARKET\_IS\_SOLVED가 현재 솔루션이 유효하다는 것을 나타낼 때까지 제한을 반복적으로 부가한다. 각 반복 동안에, 알고리즘은 COMPUTE\_MARKET\_PARAMETERS를 호출하여 시장을 특징짓는 수학식 7-12의 양을 계산한다.

도 2는 절차 SOLVE\_MARKET의 하나의 실시예를 예시한 흐름도이다. 절차는 블록 200에서 시작한다. 블록 202에서, 절차 INITIALIZE\_CONSTRAINTS가 호출된다. 이 절차의 하나의 실시예는 도 3에 도시되어 있다. 이러한 절차는 시장에서의 광고주 제한을 초기화한다. 다음으로 블록 204에서 루프에 들어간다. 블록 206에서, 절차 COMPUTE\_MARKET\_PARAMETER가 호출된다. 이 절차는 시장을 구성하는 다양한 양들을 계산한다. 이러한 절차의 하나의 실시예는 도 4에 도시되어 있다. 또한, 블록 206에서, 절차 ADD\_CONSTRAINT가 호출된다. 이 절차는 가장 중요한 트래픽 또는 제로 상수를 시장에 부가한다. 이러한 절차의 하나의 실시예는 도 6에 도시되어 있다. 블록 208에서, 절차 MARKET\_IS\_SOLVED에 의해 리턴된 값이 테스트된다. 이러한 절차의 하나의 실시예는 도 8에 도시되어 있다. 이 절차는 모든 시장의 제로 및 트래픽 제한이 해결되어 있는지를 결정한다. 이 절차가 참 값을 리턴하지 않으면, 루프 동작이 계속된다. 그렇지 않으면, 절차 SOLVE\_MARKET은 블록 210에서 종료한다.

도 3은 시장 제한을 초기화하는 알고리즘을 도시하고 있다. 알고리즘은 각 광고주의 제한을 NONE로 설정하는 루프이다. 절차 INITIALIZE\_CONSTRAINTS는 블록 300에서 시작한다. 루프는 블록 302에서 들어간다. 고려중인 각 광고주에 대해, 블록 304에서, 광고주에 의해 인덱스되는 어레이 Constraint의 값이 값 NONE로 초기화된다. 블록 306에서 모든 광고주들이 처리될 때까지 루프가 계속된다. 절차는 블록 308에서 종료한다.

도 4는 수학식 7-12에 의해 주어지는 다양한 시장 파라미터를 계산하는 방법의 하나의 실시예를 도시하고 있다. 방법은 블록 400에서 시작한다. 블록 402에서, 절차 COMPUTE\_SUMS가 호출된다. COMPUTE\_SUMS는 예비 양 FREE\_SPACES, FREE\_ADVERTISERS 및 Z, 즉 수학식 10D에서 주어지는 조화 평균값의 분모를 계산한다. 이들 양들은 나머지 공식에서 나타난다. 방법은 수학식 12에 의해 주어지는 V\_L에 대한 공식을 이용하지만, 일부 광고주들이 수동으로 그 지출을 조정한다면, 더 일반적인 수학식 16을 동일하게 잘 이용한다. 절차 COMPUTE\_SUMS의 하나의 실시예는 도 5에 도시되어 있다. 블록 404에서, V\_Bar의 값이 계산되고 블록 406에서 값 V\_L이 계산된다. 블록 408에서, 변수 TOTAL\_MARKET\_SPEND의 값이 정의된다. 방법은 블록 410에서 종료한다.

도 5는 FREE\_ADVERTISERS, FREE\_SPACES 및 Z를 계산하는 절차의 하나의 실시예를 도시하고 있다. FREE\_ADVERTISERS는 제한되지 않은 광고주의 개수를 카운트한다. FREE\_SPACES는 트래픽 제한된 광고주에 의해 차지되지 않은 검색 결과 페이지 상의 자유로운 공간의 개수를 카운트한다. Z는 각 자유로운 광고주의 역수 시장 값의 합이다. 절차는 모든 광고주들에 대해 루프를 이용하여 이들 양들을 계산한다.

절차는 블록 500에서 시작한다. 블록 503에서, FREE\_ADVERTISERS, FREE\_SPACES 및 Z가 초기화된다. 블록 504에서, 광고주를 루프 인덱스로서 이용하여 루프가 들어간다. 블록 506에서, 광고주에 대한 제한이 NONE과 동일한지가 결정된다. 그렇다면, 블록 508에서, FREE\_ADVERTISERS 및 Z의 값이 증가된다. 그리고 나서, 제어는 블록 514로 진행한다. 그렇지 않으면, 블록 510에서 광고주에 대한 Constraint의 값이 Traffic과 동일하다면, 블록 512에서 Free\_Space의 값이 감소된다. 블록 514에서, 모든 광고주들이 처리될 때까지 루프 동작이 계속된다. 절차는 블록 516에서 종료한다.

도 6은 가장 중요한 트래픽 또는 제로 제한을 CONSTRAINT[A,M] 값에 부가하는 알고리즘을 도시하고 있다. 시장이 해결되어 있다면, 할 것이 아무것도 없다. 그렇지 않으면, 알고리즘은 EXCESS\_SPEND의 부호에 따라 제로 제한 또는 트래픽 제한 중 하나를 부가한다. 과도 지출이 음이라면, 알고리즘은 가장 작은 시장 값을 가지는 자유로운 광고주에 대해 제로 제한을 부가한다. 과도 지출이 제로 또는 양이라면, 알고리즘은 가장 큰 시장값을 가지는 자유로운 광고주에 대해 트래픽 제한을 부가한다.

절차는 블록 600에서 시작한다. 블록 602에서, 절차는 절차 MARKET\_SOLVED를 호출함으로써 시장이 해결되어 있는지를 테스트한다. 이러한 절차의 하나의 실시예는 도 8에 도시되어 있다. 시장이 해결되면, 절차는 중요하고 제어는 호출 프로세스로 리턴된다. 그렇지 않으면, 블록 604에서, 절차 EXCESS\_SPEND가 호출된다. 이 절차의 하나의 실시예는 도 7에 도시되어 있다. EXCESS\_SPEND에 의해 리턴된 값이 제로보다 작으면, 블록 606에서, 알고리즘은 가장 작은 시장 값을 가지는 자유로운 광고주 A에 대해 제로 제한을 부가한다. EXCESS\_SPEND에 의해 리턴된 값이 제로 또는 양이라면, 블록 608에서 알고리즘은 가장 큰 시장값을 가지는 자유로운 광고주 A에 대해 트래픽 제한을 부가한다. 절차는 블록 610에서 종료한다.

도 7은 상기 수학식 13에 의해 주어지는 전체 부호가 있는 과도 지출을 계산하는 알고리즘을 도시하고 있다. 이는 과도 경비의 실행 전체를 유저하고, V\_L보다 작거나 V\_U보다 큰 값을 가지는 모든 광고주들로부터 액수에 부가한다. 알고리즘은 제로 제한되거나 트래픽 제한된 임의의 광고주를 무시한다.

절차는 블록 700에서 시작한다. 블록 702에서, 변수 EXCESS\_SPEND의 값이 0으로 초기화된다. 루프 동작은 블록 704에서 광고주 A를 루프 인덱스로 이용하여 개시된다. 루프에서, 블록 706에서, 현재 인덱스된 광고주 A에 대한 값이 V\_L, 최저 가능한 자유 값보다 작은지가 결정된다. 그렇다면, 블록 708에서, 광고주 A에 대한 제한이 없다면, 변수 EXCESS\_SPEND가 도 7, 블록 708에 도시된 값만큼 증가된다. 그리고 나서, 제어는 블록 714로 진행한다. 블록 710에서, 현재 인덱스된 광고주 A에 대한 값이 V\_U, 최대 가능한 자유 값보다 크다면, 제어는 블록 712로 진행한다. 광고주 A에 대한 제한이 없다면, EXCESS\_SPEND의 값이 도 7, 블록 712에 도시된 값만큼 증가된다. 그리고 나서, 제어는 블록 714로 진행한다. 루프 동작은 모든 광고주들이 처리될 때까지 반복된다. 절차는 블록 716에서 종료한다.

도 8은 현재 CONSTRAINT[A,M] 값이 유효한 솔루션인지 여부를 결정하는 알고리즘을 도시하고 있다. 각 자유로운 광고주에 대해, 알고리즘은 그 값이 최저 가능 자유값 V\_L보다 작은지 또는 최대 가능 자유값 V\_U보다 큰지를 알기 위해 체크한다. 현재 솔루션은 모든 자유 광고주들이 이를 한계내에 들어가는 경우에만 유효하다.

절차는 블록 800에서 시작한다. 블록 802에서, 루프 동작은 광고주 A를 인덱스로서 이용하여 개시된다. 블록 804에서, 광고주 A에 대한 값이 V\_L에 비교된다. 값이 V\_L보다 작으면, 블록 806에서, 광고주 A에 대한 제한이 없는 경우에, 값 NO가 절차에 의해 리턴된다. 제어는 블록 812로 진행한다. 광고주 A에 대한 값이 블록 804에서 V\_L보다 작지 않으면, 블록 808에서 이 값은 V\_U에 대해 테스트된다. V\_U를 초과하면, 블록 810에서, 광고주 A에 대해 제한이 없는 경우에, 절차는 값 NO를 리턴한다. 제어는 블록 812로 진행한다. 추가 광고주가 남아있다면, 루프 동작은 블록 802에서 계속된다. 모든 광고주들이 절차에 대해 값 NO를 리턴하지 않고 처리되었다면, 제어는 루프를 나와, 블록 814에서 절차는 값 YES를 리턴하고, 이는 시장이 해결되었다는 것을 나타낸다. 절차는 블록 816에서 종료한다.

도 9는 예산 스케일 인자를 초기화하는 알고리즘을 도시하고 있다. 알고리즘은 각 광고주의 예산 스케일 인자를 1로 설정하는 루프이다. 절차는 SOLVE 알고리즘의 시작 시에 한번 호출된다.

절차는 블록 900에서 시작한다. 블록 902에서, 루핑 동작은 광고주 A를 루핑 인덱스로서 이용하여 들어간다. 블록 904에서, 광고주에 대한 예산 스케일 인자 LAMBDA가 1의 값으로 초기화된다. 블록 906에서, 루핑 동작은 모든 광고주들이 처리될 때까지 계속된다. 절차는 블록 908에서 종료한다.

도 10-15는 예산 스케일 인자 LAMBDA[A]에 대해 푸는 단계들을 도시하고 있다. 상위 레벨 알고리즘은 도 10에 도시되어 있다. 알고리즘은 그 예산 한계에 있는 광고주의 워킹 세트를 계산함으로써 시작한다. 그리고 나서, 알고리즘은 변수 벡터를 생성하여 수학식 해결자에게 워킹 세트에서 각 광고주에 대해 하나의 변수로 캐스한다. 각 변수는 연관된 상한 및 하한을 가지고 있다. SET\_TO\_ZERO 함수는 각 광고주의 전체 지출이 정확하게 그의 예산과 맞춰하도록 LAMDBA[A]를 조정하는 외부 수학식 해결자이다. 임의의 수학식 해결자는 경계 제한을 가진 비선형 수학식의 시스템을 풀 수 있는 한 작용한다. 수학식 해결자로의 입력은 벡터 목적 함수 BUDGET\_ERROR, 변수의 개수 N, 조정하는 변수 벡터, 및 경계 제한이다. 실행함에 따라, SET\_TO\_ZERO는 BUDGET\_ERROR(I)를 호출하여 목적 함수의 I번째 성분을 평가한다. 솔루션을 구하면, 모든 I에 대해 제로와 동일한 BUDGET\_ERROR(I)로 종료한다. 솔루션이 없다면, 그 상한에서 하나 이상의 LAMBDA[A]로 종료하고, 알고리즘은 다음 반복시 워킹 세트에서 대응하는 광고주를 제거한다.

도 10은 예산 스케일 인자에 대해 해결하는 절차 SOLVE\_BUDGETS의 하나의 실시예를 예시하고 있다. 절차는 블록 1000에서 시작한다. 블록 1002에서, 변수 N은 0으로 초기화되고, 변수 WORKING\_SET는 절차 BUDGET\_LIMITED\_ADVERTISERS의 결과와 동일하게 설정된다. 이 절차의 하나의 실시예는 도 12에 도시되어 있다. 블록 1004에서, 루핑 동작은 광고주 A를 루핑 인덱스로서 이용하여 개시된다. 블록 1006에서, 변수 N이 증가되고, 변수 N에 의해 인덱스된 벡터 VARIABLES의 엔트리가 현재 광고주에 대한 벡터 LAMBDA의 엔트리로의 참조와 동일하게 설정된다. 절차 MIN\_LAMBDA가 호출되어 변수 LOWER\_BOUND에 대한 값을 결정한다. MIN\_LAMBDA는 광고주가 어느 곳에서든지 제로 제한되기 이전에 현재 인덱스된 광고주의 최소 람다를 계산한다. 이러한 절차의 하나의 실시예는 도 13에 도시되어 있다. 그리고 나서, 절차 MAX\_LAMBDA는 변수 UPPER\_BOUND에 대한 값을 결정하는데 이용된다. MAX\_LAMBDA는 광고주가 어느 곳에서든지 트래픽 제한되기 이전에 현재 인덱스된 광고주의 최대 람다를 계산한다. 이 절차의 하나의 실시예는 도 14에 도시되어 있다. 블록 1008에서, 루핑 동작은 모든 광고주 A가 처리될 때까지 계속된다.

루핑 동작을 완료한 후, 블록 1010에서, 절차 SET\_TO\_ZERO가 호출되어 상기 설명된 바와 같이 입력, 함수 BUDGET\_ERROR에 의해 정의된 수학식 시스템을 푼다. 이러한 함수의 하나의 실시예는 도 11에 도시되어 있다. SOLVE\_BUDGETS 절차는 블록 1012에서 종료한다.

도 11은 워킹 세트의 광고주가 예산 초과하거나 미만인 액수를 계산하는 절차를 도시하고 있다. 절차는 워킹 세트로부터 I번째 광고주를 검색하고, 그 현재 전체 지출과 그 예산 한계 간의 차이를 리턴한다.

절차 BUDGET\_ERROR는 블록 1100에서 시작한다. 블록 1102에서, I번째 광고주가 검색된다. 블록 1104에서, BUDGET\_ERROR의 값이 절차 TOTAL\_SPEND에 의해 리턴된 결과와 광고주 A에 대한 BUDGET\_LIMIT간의 차이로서 계산된다. 절차 TOTAL\_SPEND의 하나의 실시예가 도 16에 도시되어 있다. 절차는 블록 1106에서 종료한다.

도 12는 그 예산 한계에 있는 광고주의 워킹 세트를 계산하는 절차의 하나의 실시예를 도시하고 있다. 워킹 세트는 그 예산을 초과하거나 최대 가능 값보다 작은 LAMBDA[A]를 가지는 모든 광고주들을 포함한다. 알고리즘은 이를 2가지 예산 한계 조건 중 하나를 만족시키는 모든 광고주를 워킹 세트에 부가한다.

절차는 블록 1200에서 시작한다. 블록 1202에서, 변수 BUDGET\_LIMITED\_ADVERTISERS가 초기화된다. 루핑 동작은 블록 1204에서 광고주 A를 루핑 인덱스로서 이용하여 초기화된다. 블록 1206에서, 광고주 A에 대한 절차 TOTAL\_SPEND의 현재 값이 광고주 A에 대한 예산을 초과하는지가 결정된다. 절차 TOTAL\_SPEND의 하나의 실시예가 도 16에 도시되어 있다. 블록 1206에서 테스트된 조건이 참이면, 변수 BUDGET\_LIMITED\_ADVERTISERS의 값이 블록 1208에서 광고주 A에 대한 예산만큼 증가된다. 그렇지 않다면, 블록 1210에서, 현재 인덱스된 광고주에 대한 람다 값이 광고주 A에 대한 MAX\_LAMBDA의 현재 값보다 적은지를 결정한다. 절차 MAX\_LAMBDA의 하나의 실시예가 도 14에 도시되어 있다. 그렇다면, 블록 1212에서, 변수 BUDGET\_LIMITED\_ADVERTISERS의 값이 광고주 A에 대한 예산만큼 증가된다. 그렇지 않으면, 제어는 블록 1214로 진행한다.

블록 1214에서, 추가 광고주가 남아있는 경우, 제어는 블록 1204로 리턴한다. 그렇지 않으면, 모든 광고주가 처리되었다면, 절차는 블록 1216에서 종료한다.

도 13은 광고주의 최적 지출이 모든 시장에서 읊이 되기 전에 광고주가 가질 수 있는 최소 LAMBDA[A]를 계산하는 절차의 하나의 실시예를 도시하고 있다. 각 시장에 대해, 알고리즘은 수학식 15를 이용하여 광고주의 최소 LAMBDA[A]를 계산한다. 모든 시장에 대해 최소는 임의의 시장내의 최소값이다.

절차는 블록 1300에서 시작한다. 블록 1302에서, 변수 MIN\_LAMBDA의 값이 1로 초기화된다. 루핑 동작은 블록 1304에서 시장 M을 루핑 인덱스로서 이용하여 들어간다. 블록 1306, 1308에서, 상기 수학식 15가 구현되어 광고주에 대한 최소 람다를 결정한다. 블록 1310에서, 처리될 추가 시장이 남아있는 경우, 제어는 블록 1304로 리턴한다. 그렇지 않으면, 절차는 블록 1312에서 종료한다.

도 14는 광고주가 모든 시장에서 트래픽 한계에 도달하기 이전에 광고주가 가질 수 있는 최대 LAMBDA[A]를 계산하는 하나의 절차를 도시하고 있다. 각 시장에 대해, 알고리즘은 수학식 15를 이용하여 광고주의 최대 LAMBDA[A]를 계산한다. 모든 시장에 대한 최대는 임의의 시장내의 최대값이다.

절차는 블록 1400에서 시작한다. 블록 1402에서, 변수 MAX\_LAMBDA는 0으로 초기화된다. 루프 동작은 블록 1404에서 시장 M을 루프 인덱스로 이용하여 시작한다. 블록 1406, 1408에서, 상기 수학식 15가 구현되어 광고주에 대한 최대 람다를 결정한다. 루프는 모든 시장이 처리될 때까지 블록 1410에서 계속된다. 절차는 블록 1412에서 종료한다.

도 15는 임의의 예산 제한 위반이 있는지를 결정하는 절차 BUDGET\_ARE\_SOLVED의 하나의 실시예를 예시하고 있다. 알고리즘은 모든 광고주가 그 예산을 초과하지 않고 LAMBDA[A]의 최대 가능값을 가지는 것을 보장하도록 체크한다. 알고리즘은 모든 광고주들이 이 조건을 만족할 때에만 YES를 리턴한다.

절차는 블록 1502에서 시작한다. 루프는 블록 1504에서 각 광고주 A에 대해 시작한다. 블록 1506에서, 광고주 A에 대한 절차 TOTAL\_SPEND에 의해 리턴된 결과는 광고주 A의 예산과 비교된다. 절차 TOTAL\_SPEND의 하나의 실시예는 도 16에 도시되어 있다. 결과가 예산보다 크다면, 절차는 블록 1508에서 값 NO를 리턴한다. 그렇지 않으면, 블록 1510에서, 절차 TOTAL\_SPEND에 의해 리턴된 결과가 광고주에 대한 예산보다 적다면, 단계 1512에서, 광고주에 대한 람다가 광고주에 대한 절차 MAX\_LAMBDA에 의해 리턴된 값보다 작은 경우에, 값 NO가 절차에 의해 리턴된다. 절차 MAX\_LAMBDA의 하나의 실시예는 도 15에 도시되어 있다. 블록 1512에서, 모든 광고주들이 처리될 때까지 루프가 계속된다. 어떤 광고주도 루프를 통한 반복 동안에 값 NO를 리턴하지 않는다면, 블록 1514에서, 값 YES가 리턴되고, 절차는 블록 1516에서 종료한다.

도 16은 전체 시장에 대해 광고주의 전체 최적 지출을 계산하는 절차 TOTAL\_SPEND의 하나의 실시예를 도시하고 있다. 이는 OPTIMAL\_SPEND(A,M)을 이용하여 각 시장에서의 액수를 계산하고 결과를 실행 전체에서 합산한다.

절차는 블록 1600에서 시작한다. 블록 1602에서, 변수 TOTAL\_SPEND의 값이 0으로 초기화된다. 각 시장 M에 대해 블록 1604, 1606, 1608을 포함하는 루프에서, 변수 TOTAL\_SPEND의 값이 절차 OPTIMAL\_SPEND에 의해 리턴된 결과에 의해 증가된다. 이러한 절차의 하나의 실시예가 도 19에 도시되어 있다. 모든 시장들이 처리된 후, 절차는 블록 1610에서 종료한다.

도 17은 현재의 CONSTRAINT[A,M] 및 LAMBDA[A] 값들이 모든 제로, 트래픽 및 예산 한계를 만족하는지 여부를 결정하는 절차 IS\_SOLVED의 하나의 실시예를 도시하고 있다. 알고리즘은 우선 모든 시장이 제로 및 트래픽 제한을 만족하는 것을 보장하도록 체크한다. 그리고 나서, 알고리즘은 모든 광고주가 그 예산 한계를 만족시키는 것을 보장하도록 체크한다. 알고리즘은 모든 조건들이 참인 경우에만 YES를 리턴한다.

절차는 블록 1700에서 시작한다. 블록 1702에서, 시장 M을 루프 변수로서 이용하는 루프가 들어간다. 블록 1704에서, 절차 MARKET\_IS\_SOLVED에 의해 리턴된 값이 테스트된다. 절차 MARKET\_IS\_SOLVED의 하나의 실시예가 도 8에 도시되어 있다. 절차가 음의 값을 리턴한다면 절차 IS\_SOLVED는 값 NO를 리턴한다. 그렇지 않으면, 블록 1706에서, 루프는 계속해서 다른 시장 M을 테스트한다. 일단 모든 시장이 테스트되면, 절차 BUDGETS\_ARE\_SOLVED에 의해 리턴된 값이 테스트된다. 절차 BUDGETS\_ARE\_SOLVED의 하나의 실시예가 도 15에 도시되어 있다. 이러한 절차가 양의 값을 리턴하지 않는다면, 절차 IS\_SOLVED는 단계 1708에서 값 NO를 리턴한다. 그렇지 않으면, 블록 1710에서, 절차는 값 YES를 리턴한다. 절차는 블록 1712에서 종료한다.

도 18은 각 시장에서 각 광고주에 대한 최종 최적 지출 액수를 기록하는 절차 COMPUTE\_SPENDING의 하나의 실시예를 도시하고 있다. 알고리즘은 각 SPEND[A,M] 값을 OPTIMAL\_SPEND(A,M)의 현재 값으로 설정하는 루프이다.

절차는 블록 1700에서 시작한다. 블록 1702에서, 외부 루프는 광고주 A를 루프 변수로서 이용함으로써 들어간다. 블록 1704에서, 내부 루프는 시장 M을 루프 변수로서 이용함으로써 들어간다. 블록 1806에서, 어레이 SPEND의 엔트리는 절차 OPTIMAL\_SPEND에 의해 리턴되는 현재 값을로 설정된다. 이러한 절차의 하나의 실시예는 도 19에 도시되어 있다. 모든 시장들이 광고주 A의 값에 대해 처리된 후, 회부 루프에 대한 루프 변수로서의 광고주의 값이 증가된다. 모든 광고주들이 처리된 후, 절차는 블록 1812에서 종료한다.

도 19는 수학식 7을 이용하여 단일 시장에서 광고주의 최적 지출을 계산하는 절차 OPTIMAL\_SPEND의 하나의 실시예를 도시하고 있다. 값이 제로보다 작거나 트래픽 한계보다 큰 경우에, 알고리즘은 이를 적절한 값으로 제한한다.

절차는 블록 1900에서 시작한다. 블록 1902에서, 변수 OPTIMAL\_SPEND에 대한 값이 시장에서 광고주에 대한 TOTAL\_SPEND에 기초하여 결정된다. 이러한 동작에 대한 절차 TOTAL\_SPEND의 하나의 실시예가 도 16에 도시되어 있다. 블록 1904에서, OPTIMAL\_SPEND의 결과는 OPTIMAL\_SPEND와 0 중 큰 것 및 OPTIMAL\_SPEND와 TRAFFIC\_LIMIT 중 최소로 설정된다. 절차는 블록 1906에서 종료한다.

도 20은 광고주의 시장 값을 계산하는 절차의 하나의 실시예를 도시하고 있다. 절차는 블록 2000에서 시작한다. 값 VALUE는 광고주의 예산 스케일 인자 람다, 그의 클릭당 이익, 및 그 클릭스루 레이트의 곱이다. 클릭당 이익 및 클릭스루 레이트는 임의의 간편한 소스로부터 얻어질 수 있다.

그러므로, 도 22의 블록도는 모두가 네트워크(22100)에 접속되는, 복수의 클라이언트 컴퓨터(2202), 복수의 광고주 웹 서버(2204), 어카운트 관리 서버(2206), 및 검색 엔진 웹 서버(2208)를 포함하는 분산 시스템(2200)을 도시하고 있다. 네트워크(2210)는 이하에서는 일반적으로 인터넷으로 지칭된다. 개시된 시스템 및 방법이 인터넷에 특별히 유용하지만, 클라이언트 컴퓨터(2202), 광고주 웹 서버(2204), 어카운트 관리 서버(2206), 및 검색 엔진 웹 서버(2208)는 다수의 다른 타입의 네트워크들 중 하나를 통해 함께 접속될 수 있다는 것은 자명하다. 그러한 네트워크는 로컬 영역 네트워크(LAN), 다른 외이드 영역 네트워크(WAN), 및 상용 정보 서비스와 같이 전화선을 통해 액세스되는 지역 네트워크를 포함한다. 클라이언트 및 서버 프로세스들은 하나의 컴퓨터 상에서 동시에 실행하는 다른 프로그램들을 포함할 수 있다.

클라이언트 컴퓨터(2202)는 종래의 퍼스널 컴퓨터(PC), 워크스테이션, 또는 임의의 다른 크기의 컴퓨터 시스템일 수 있다. 각 클라이언트(2202)는 통상 하나 이상의 프로세서, 메모리, 입출력 디바이스, 및 종래 모뎀과 같은 네트워크 인터페이스를 포함한다. 광고주 웹 서버(2204), 어카운트 관리 서버(2206) 및 검색 엔진 웹 서버(2208)는 동일하게 구성될 수 있다. 그러나, 광고주 웹 서버(2204), 어카운트 관리 서버(2206) 및 검색 엔진 웹 서버(2208)는 각각 분리된 사설 네트워크에 의해 접속된 다수의 컴퓨터를 포함할 수 있다. 실제로, 네트워크(2210)는 수십만 개의 개별적인 컴퓨터 네트워크를 포함할 수 있다.

클라이언트 컴퓨터(2202)는 NAVIGATOR, EXPLORER, 또는 MOSAIC 브라우저 프로그램과 같은 웹 브라우저 프로그램(2212)을 실행하여, 광고주 서버(2204)에 저장된 웹 페이지 또는 기록(2214)을 로케이팅 할 수 있다. 브라우저 프로그램(2212)은 사용자가 검색될 특정 웹 페이지(2214)의 어드레스를 입력할 수 있게 한다. 이들 어드레스들은 유니폼 리소스 로케이터 또는 URL로서 지정된다. 뿐만 아니라, 일단 하나의 페이지가 검색되었으면, 브라우저 프로그램(2212)은 사용자가 다른 웹 페이지로의 하이퍼링크에 "클릭한" 경우에 다른 페이지 또는 기록으로 액세스를 제공할 수 있다. 그러한 하이퍼링크는 웹 페이지(2214)내에 배치되고, 사용자가 다른 페이지의 URL을 입력하고 그 페이지를 검색하는 자동화된 방법을 제공한다. 페이지는 컨텐트 보통 텍스트 정보, 또는 소프트웨어 프로그램, 그래픽스, 오디오 신호, 비디오 등과 같은 더 복잡한 디지털로 인코딩된 멀티미디어 컨텐트로서 포함하는 데이터 기록일 수 있다.

하나의 실시예에서, 클라이언트 컴퓨터(2202)는, FTP, SNMP, TELNET 및 본 기술분야에서 주지된 다수의 다른 프로토콜이 이용될 수 있지만, 하이퍼텍스트 전달 프로토콜(HTTP)에 의해 제공되는 기능을 이용하여, 어카운트 관리 서버(2206), 검색 엔진 서버(2208) 및 광고주 서버(2204)를 포함하는 다양한 네트워크 정보 제공자와 네트워크(2210)를 통해 통신한다. 양호하게는, 검색 엔진 서버(2208), 어카운트 관리 서버(2206), 및 광고주 서버(2204)는 월드 와이드 웹 상에 배치된다.

상기 설명된 바와 같이, 적어도 2가지 타이브이 서버가 실시예에서 고려된다. 고려되는 제1 서버는 컴퓨터 저장 매체(2220) 및 차리 시스템(2222)을 포함하는 어카운트 관리 서버(2206)이다. 데이터베이스(2224)는 어카운트 관리 서버(2206)의 저장 매체(2220) 상에 저장된다. 데이터베이스(2224)는 하나의 실시예에서의 광고주 가입 어카운트 정보를 포함하는 광고주 어카운트 정보를 포함한다. 여기에 개시된 시스템 및 방법은 어카운트 관리 서버(2206) 상의 메모리나 대량 저장 디바이스와 같은 컴퓨터 저장 매체 상의 실행 가능한 명령으로서 저장된 소프트웨어로 구현된다는 것은 상기 설명으로부터 잘 이해될 것이다. 클라이언트 컴퓨터(2202) 상에서 실행되는 종래 브라우저 프로그램(2212)은 어카운트 관리 서버(2206)에 저장된 광고주 어카운트 정보에 액세스하는데 이용될 수 있다. 바람직하게는, 어카운트 관리 서버(2206)로의 액세스는 어카운트 관리 및 검색 결과 배치 프로그램 및 어카운트 정보를 외부 변경으로부터 보호하는 도시되지 않은 방화벽을 통해 달성될 수 있다. 추가적인 보안은 Secure HTTP 또는 Secure Sockets Layer와 같은 표준 통신 프로토콜에 대한 향상을 통해 제공된다.

고려되는 제2 서버 탑재은 검색 엔진 웹 서버(2208)이다. 검색 엔진 프로그램은, 검색 엔진 웹 서버 URL 또는 브라우저 프로그램(2212)을 통해 검색 엔진 웹 서버(2208)에 질의를 제출할 수 있는 다른 웹 서버 상의 사이트에 네비게이션 할 때, 네트워크 사용자 또는 검색자들이 키워드 질의를 타이핑하여, 월드 와이드 웹 상에서 가용한 수백만개의 페이지들 중에서 관심있는 페이지들을 식별할 수 있게 한다. 하나의 실시예에서, 검색 엔진 웹 서버(2208)는 어카운트 관리 서버(2206)에 의해 수행된 입찰 프로세스의 결과에 의해 얻어지고 포맷되는 관련된 엔트리를 적어도 부분적으로 포함하는 검색 결과 리스트를 생성한다. 검색 엔진 웹 서버(2208)는 클라이언트 컴퓨터(2202)에서 사용자에 의해 입력된 검색어와 관련된 정보를 포함하는 문서로의 하이퍼텍스트 링크의 리스트를 생성한다. 검색 엔진 웹 서버는 이 리스트를 웹 페이지의 형태로 네트워크 사용자에게 송신하고, 이는 클라이언트 컴퓨터(2202)에서 실행되는 브라우저(2212) 상에 표시된다. 검색 엔진 웹 서버의 하나의 실시 예에는 URL <http://www.overture.com>의 웹 페이지를 네비게이션 함으로써 발견될 수 있다.

검색 엔진 웹 서버(2208)는 인터넷(2210)에 접속된다. 하나의 실시예에서, 검색 엔진 웹 서버(2208)는 사용자 질의에 응답하여 검색 결과를 생성하는데 이용되는 검색 리스트 기록을 포함하는 검색 데이터베이스(230)를 포함한다. 뿐만 아니라, 검색 엔진 웹 서버(2208)는 어카운트 관리 서버(2206)에 접속될 수 있다. 어카운트 관리 서버(2206)는 인터넷에 접속된다. 검색 엔진 웹 서버(2208) 및 어카운트 관리 서버(2206)는 클라이언트 컴퓨터(2202)에 배치된 사용자들의 다른 정보 요구를 충족시킨다.

클라이언트 컴퓨터(2202)에 위치한 한 부류의 사용자들은 광고하는 웹 사이트 프로모터 또는 광고주 웹 서버(2204) 상에 위치한 광고주 웹 페이지(2214)를 가지는 광고주들과 같은 네트워크 정보 제공자일 수 있다. 이들 광고하는 웹 사이트 프로모터 또는 광고주들은 어카운트 관리 서버(2206) 상의 저장 장치(2220)에 상주하는 어카운트 정보에 액세스하기 원할 수 있다. 광고하는 웹 사이트 프로모터는 어카운트 관리 서버(2206)에 상주하는 어카운트를 통해, 다른 광고주들과의 경쟁적인 입찰 프로세스에 참여한다. 광고주는 광고주의 웹 사이트의 내용과 관련된 임의의 개수의 검색어에 대해 입찰할 수 있다. 하나의 실시예에서, 대응하는 웹 사이트에 대한 검색 리스트에서 입찰되는 검색어의 관련성은 어카운트 관리 서버(2206)의 프로세서(2222)에서 실행되는 컴퓨터 프로그램을 이용하여 평가되고, 여기에서 컴퓨터 프로그램은 선정된 편집 규칙 세트에 따라 검색어 및 대응하는 웹 사이트를 평가한다.

광고주에 의한 입찰되는 검색어를 이용한 검색이 실행된 경우에, 검색 엔진(2208)에 의해 생성되는 검색 결과 리스트 페이지 상에서 더 높은 입찰가는 더 양호한 위치를 받는다. 하나의 실시예에서, 광고주에 의해 입찰되는 액수는 광고주의 웹사이트가 검색 결과 리스트 페이지 상의 하이퍼링크를 통해 액세스될 때마다 광고주의 어카운트로부터 인출되는 금액으로 구성된다. 다른 실시예에서, 광고주의 가입 어카운트는 검색 질의에 응답하여 광고주의 검색 리스트가 검색자에게 서빙되거나 표시될 때마다 소정 액수만큼 인출된다. 검색자는 컴퓨터 입력 디바이스로 하이퍼링크를 "클릭"하여 검색 요구를 개시시키고 광고주의 하이퍼링크와 연관된 정보를 검색한다. 양호하게는, 각 액세스 또는 검색 결과 리스트 하이퍼링크상의 "클릭"은 검색 엔진 웹 서버(2208)에 재지향되어 "클릭"을 광고주에 대한 어카운트 식별자와 연관시킨다. 검색자에게 명백하지 않은 이러한 재지향 액션은, 검색자에 의해 클릭된 검색 결과 리스트 하이퍼링크를 이용하여 광고주의 URL에 액세스하기 이전에, 검색 결과 페이지로 코딩된 어카운트 식별 정보를 액세스할 것이다. 다른 실시예에서, 광고주의 가입 어카운트로부터 소정 액수의 인출을 유발하는 것은 광고주의 URL로의 이러한 클릭스루 동작이다. 어카운트 식별 정보는 검색 요구 이벤트로서의 검색 요구로부터의 정보와 함

께 광고주의 어카운트에 기록된다. 이러한 메커니즘을 통해 얻어지는 정보는 본 기술분야에서 주지된 종래 서버 시스템 또는 로그를 이용하여서는 불가능한 방식으로 URL을 가지는 어카운트 식별자와 결론적으로 매칭하므로, 정확한 어카운트 지불 기록이 유지된다. 일부 실시예들에서, 검색 결과 리스트 페이지 상의 광고주의 웹 사이트 기술 및 하이퍼링크는 광고주의 리스트가 유료 리스트라는 표시를 동반한다.

클라이언트 컴퓨터(2202)에서의 제2 부류의 사용자들은 웹 상에서 특정 정보를 찾는 검색자들로 구성된다. 검색자들은 그 브라우저(2212)를 통해 웹 서버(2208)에 상주하는 검색 엔진 웹 페이지(2232)에 액세스한다. 검색 엔진 웹 페이지(2232)는 검색자가 하나 이상의 키워드를 포함하는 검색어를 태이핑하는 질의 박스를 포함한다. 다른 게는, 검색자는 검색 엔진 웹 서버(2208)에 하이퍼링크되고 원격 웹 서버에 저장된 웹 페이지에 위치한 질의 박스를 통해 검색 엔진 웹 서버(2208)에 질의한다. 검색자가 검색어를 입력하는 것을 종료한 경우에, 검색자는 제공된 하이퍼링크를 클릭함으로써 질의를 검색 엔진 웹 서버(2208)에 송신 할 수 있다. 검색 엔진 웹 서버(2208)는 검색 결과 리스트 페이지를 생성하고 이 페이지를 클라이언트 컴퓨터(2202)의 검색자에게 송신한다.

검색자는 검색 결과 페이지 상의 각 리스트와 연관된 하이퍼텍스트 링크를 클릭하여 대응하는 웹 페이지에 액세스한다. 하이퍼텍스트 링크는 인터넷 상의 어디에서나 웹 페이지를 액세스하고, 광고주 웹 서버(2204) 상에 위치한 광고주 웹 페이지(2214)로의 유료 리스트를 포함한다. 하나의 실시예에서, 검색 결과 리스트는 광고주 입찰의 결과로서 배치되지 않고 INKTOMI, LYCOS 또는 YAHOO! 검색 엔진과 같은 종래 월드 와이드 웹 검색 엔진에 의해 생성된 무료 리스트를 포함한다. 무료 하이퍼텍스트 링크는 편집 팀에 의해 데이터베이스(2230)에 수동으로 인ベ스된 링크를 포함할 수 있다.

상기 설명으로부터, 본 실시예들은 광고주에 의해 지출되는 액수에 비례하여 광고주의 리스트를 표시하기 위한 방법 및 장치를 제공하는 것을 알 수 있다. 각 광고주는 하나의 검색어에 얼마를 지출하기를 원하는지를 결정하고, 이에 따라 검색 제공자는 광고주의 리스트를 표시한다.

본 발명의 특정 실시예가 도시되고 설명되었지만, 변경이 가능할 수 있다. 그러므로, 본 발명의 진정한 사상 및 범주에 드는 첨부된 청구의 범위는 그러한 변경 및 변형을 포함하려는 것이다.

그러므로, 상기 상세한 설명은 제한적이라기보다는 예시적인 것으로 간주되어야 하고, 이하의 청구의 범위 및 그 등 가들은 본 발명의 사상 및 범주를 정의하려는 것이라는 것은 자명하다.

#### (57) 청구의 범위

##### 청구항 1.

검색자들이 검색 질의들을 데이터베이스에 제출하고 적어도 일부 광고주-후원된 검색 리스트들(search listings)을 포함하는 검색 리스트들을 수신하는 데이터베이스 검색 시스템을 위한 영업 방법에 있어서,

지정된 가격에 대해 각 광고주에게 검색자 계약들(searcher engagements)의 지정된 양을 판매하는 단계; 및

이어서, 검색 질의들에 응답하고, 상기 검색자 계약들의 각 광고주의 지정된 양에 비례하여 각 광고주의 검색 리스트들을 제공하는 단계

를 포함하는 영업 방법.

##### 청구항 2.

제1항에 있어서, 상기 검색자 계약들의 지정된 양을 판매하는 단계는,

상기 각 광고주의 검색 리스트들의 검색자 임프레션들(searcher impressions)을 판매하는 단계를 포함하는 영업 방법.

##### 청구항 3.

제1항에 있어서, 상기 검색자 계약들의 지정된 양을 판매하는 단계는,

상기 각 광고주의 검색 리스트들의 검색자 클릭스루들(searcher clickthroughs)를 판매하는 단계를 포함하는 영업 방법.

##### 청구항 4.

제1항에 있어서, 상기 검색자 계약들의 지정된 양을 판매하는 단계는, 각 광고주의 검색 리스트의 클릭스루 이후에 상기 각 광고주의 웹 사이트에서의 포스트-클릭스루 검색자 액션들을 판매하는 단계를 포함하는 영업 방법.

##### 청구항 5.

제1항에 있어서, 상기 검색자 계약들의 지정된 양을 판매하는 단계는, 소정 기간 동안에 상기 각 광고주의 검색 리스트들의 검색자 임프레션들 및 클릭스루들을 중 하나를 판매하는 단계를 포함하는 영업 방법.

#### 청구항 6.

제1항에 있어서,

검색 요청을 수신하는 단계;

각각의 광고주에 대해, 수신된 검색자 계약들의 각 개수와 상기 각 광고주가 수신해야 하는 검색자 계약들의 각 개수와 관련된 각 차이값을 결정하는 단계; 및

상기 결정된 차이값에 따라 검색 리스트들을 리턴하는 단계

를 더 포함하는 영업 방법.

#### 청구항 7.

제1항에 있어서, 상기 검색자 계약들의 지정된 양을 판매하는 단계는,

각 광고주에 대한 클릭당 비용, 및 상기 각 광고주에 대한 추정된 클릭 회수 중 적어도 하나를 결정하는 단계를 포함하는 영업 방법.

#### 청구항 8.

제7항에 있어서, 시장에 대한 이력 데이터에 기초하여 각 광고주에 대한 클릭스루 레이트를 결정하는 단계를 더 포함하는 영업 방법.

#### 청구항 9.

제8항에 있어서, 상기 클릭스루 레이트를 결정하는 단계는,

상기 검색 리스트가 수신한 임프레션들의 개수에 대한 상기 각 광고주의 검색 리스트가 수신한 클릭 개수의 비율로서 상기 클릭스루 레이트를 추정하는 단계를 포함하는 영업 방법.

#### 청구항 10.

제1항에 있어서, 각 광고주에게 최적의 광고 비용을 추천하는 단계; 및

다른 시장들 사이에서 상기 광고주 광고 비용을 할당하는 단계를 더 포함하는 영업 방법.

#### 청구항 11.

검색자들이 검색 질의들을 데이터베이스에 제출하고 적어도 일부 광고주-후원된 검색 리스트들을 포함하는 검색 리스트들을 수신하는 데이터베이스 검색 시스템을 위한 방법을 구현하도록 구성된 컴퓨터 판독가능한 컴퓨터 코드를 저장하는 컴퓨터 판독가능한 매체에 있어서, 상기 컴퓨터 판독가능한 컴퓨터 코드는,

지정된 가격에 대해 각 광고주에게 검색자 계약들 - 상기 검색자 계약들은 검색자 임프레션들 및 검색자 클릭스루들 중 적어도 하나를 포함함 -의 지정된 양을 제공하기 위한 제1 컴퓨터 판독가능한 코드; 및

연이어 수신된 검색 질의들에 응답하고 검색자 계약들의 상기 각 광고주의 지정된 양에 비례하여 각 광고주의 검색 리스트들을 제공하기 위한 제2 컴퓨터 판독가능한 코드

를 포함하는 컴퓨터 판독가능한 저장 매체.

#### 청구항 12.

제11항에 있어서,

각 광고주에 대한 클릭스루당 비용을 결정하기 위한 제3 프로그램 코드; 및

상기 각 광고주에게 전달가능한 추정된 클릭스루의 개수를 결정하기 위한 제4 프로그램 코드

를 더 포함하는 컴퓨터 판독가능한 저장 매체.

### 청구항 13.

제12항에 있어서, 상기 제3 컴퓨터 프로그램 코드는 상기 각 광고주에 대한 추정된 전체 광고 판매 및 추정된 클릭스루 레이트의 합수로서 상기 클릭스루당 비용을 결정하도록 구성되는 컴퓨터 판독가능한 저장 매체.

### 청구항 14.

제12항에 있어서, 시장에 대한 이력 데이터에 기초하여 상기 클릭스루 레이트 및 상기 추정된 전체 광고 판매를 결정하도록 구성되는 제5 컴퓨터 프로그램 코드를 더 포함하는 컴퓨터 판독가능한 저장 매체.

### 청구항 15.

제12항에 있어서, 상기 제4 컴퓨터 프로그램 코드는 추정된 전체 광고 판매에 대한 각 광고주의 광고 비용의 비율의 합수로서 상기 추정된 클릭스루 회수를 결정하도록 구성되는 컴퓨터 판독가능한 저장 매체.

### 청구항 16.

제11항에 있어서, 상기 광고주의 전체 광고 예산, 상기 광고주의 클릭스루당 이익, 및 일부 리턴 레이트에 기초하여 각 광고주의 광고 비용을 최적화하기 위한 제3 프로그램 코드를 더 포함하는 컴퓨터 판독가능한 저장 매체.

### 청구항 17.

제16항에 있어서, 상기 각 광고주에 대한 예산 스케일 인자를 결정하기 위한 제4 컴퓨터 프로그램 코드를 더 포함하는 컴퓨터 판독가능한 저장 매체.

### 청구항 18.

제17항에 있어서, 상기 제4 컴퓨터 프로그램 코드는,

우선 모든 제로 제한(zero constraint) 및 트래픽 제한(traffic constraint)에 대해 풀고,

트래픽 한계에서 광고주들에 의해 차지되지 않는 검색 질의에 응답하여 제공되는 검색 결과들의 페이지 상에서 자유 공간의 개수, 및 현재 제로 한계 또는 트래픽 한계에 있지 않은 광고주들의 개수를 결정하며,

상기 각 광고주의 예산 제한을 결정하는 방정식들의 시스템을 풀도록 구성되는 컴퓨터 판독가능한 저장 매체.

### 청구항 19.

사용자에 의해 원격 입력 디바이스를 통해 입력되는 키워드에 응답하여 결과 리스트를 생성하기 위해 컴퓨터 네트워크 상에서 동작하는 장치에 있어서,

상기 장치는 상기 컴퓨터 네트워크에 접속된 컴퓨터 시스템을 포함하고,

각 리스트가 광고주와 연관되는 복수의 리스트를 포함하는 데이터베이스;

사용자가 키워드를 입력하는 것에 응답하여 결과 리스트를 생성하기 위한 프로그래밍 코드 - 상기 결과 리스트는 상기 사용자에 의해 입력된 상기 키워드와의 매치를 생성하는 연관된 키워드를 가지는 리스트들을 포함함 - ; 및

사용자 계약들의 개수를 지정하는 광고주 가입 주문들을 수락하고, 광고주 가입 어카운트들(accounts)을 관리하며, 광고주 리스트가 생성된 검색 리스트에 포함될 때 각 광고주의 가입 어카운트를 조정하는 것을 포함하여, 광고주 가입을 관리하기 위한 프로그래밍 코드

를 저장하는 장치.

### 청구항 20.

제19항에 있어서, 상기 컴퓨터 시스템은 각 광고주에 의해 제공되는 보상에 응답하여 상기 사용자 계약들의 개수를 저장하는 장치.

### 청구항 21.

제20항에 있어서, 상기 컴퓨터 시스템은 각 광고주에 의해 제공되는 보상에 대해 그 보답으로 제공되는 추정된 클릭 수를 개수를 결정하고, 상기 추정된 클릭스루의 개수는 모든 광고주에 의해 제공되는 전체 보상에 대한 상기 각 광고주에 의해 제공되는 상기 보상의 비율에 관련되는 장치.

### 청구항 22.

PFP(pay-for-placement) 데이터베이스 검색 시스템에 대한 가입 방법에 있어서,  
광고주들에게 지정된 비용으로 지정된 개수의 검색자 계약들을 제공하는 단계;  
하나 이상의 가입 광고주로 가입 어카운트들을 개시하는 단계;  
검색자들로부터 검색 요청들을 수신하는 단계;  
상기 검색 요청들에 응답하여, 가입 광고주들의 검색 리스트들을 포함한 검색 결과들을 제공하는 단계; 및  
상기 검색 결과들을 제공하는 것에 응답하여 상기 가입 광고주들의 상기 가입 어카운트들을 조정하는 단계  
를 포함하는 가입 방법.

### 청구항 23.

제22항에 있어서,

상기 각 광고주의 광고 예산, 클릭스루당 이익 및 외부 리턴 레이트 중 적어도 하나에 기초하여 각 광고주에게 최적의 광고주 비용 액수를 추천하는 단계를 더 포함하는 가입 방법.

### 청구항 24.

제23항에 있어서,

복수의 시장에 걸쳐 상기 광고주의 비용을 제한하기 위해 상기 각 광고주에 대한 예산 스케일 인자를 결정하는 단계  
를 더 포함하는 가입 방법.

#### 요약

광고주(2202)는 검색자(2202)로부터의 질의에 응답하여 검색 결과(2212)가 제공되는 데이터베이스 내에 그 검색 리스트를 포함하도록 지불하는 PFP 데이터베이스 검색 시스템(2200)은 광고주가 검색어에 지출하기를 원하는 금액을 결정한다. 검색 제공자(2208)는 각 광고주가 지출하는 금액에 비례하여 광고주의 리스트(2212)를 표시한다. 이것은 광고주가 데이터베이스 검색 시스템(2200)에 가입하여, 소정 기간동안의 가입에 대해 열미를 지불할지를 결정할 수 있게 한다. 검색 제공자(2208)는 광고주(2202)에 대해 최적 지출 액(2206)을 추천할 수 있다.

#### 내용도

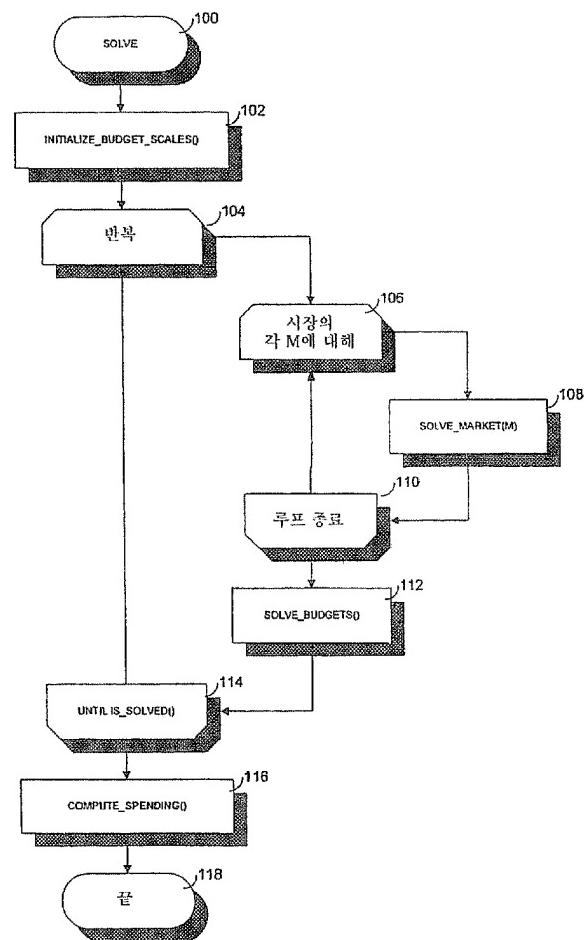
#### 도 22

#### 색인어

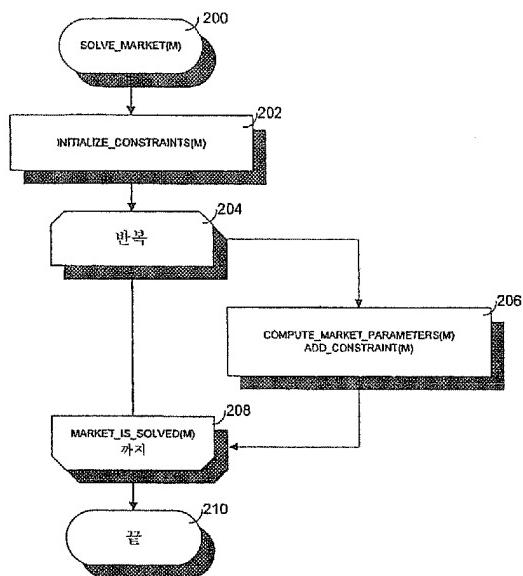
PFP 검색, 검색 엔진, 인터넷, 광고, 영업방법

#### 도면

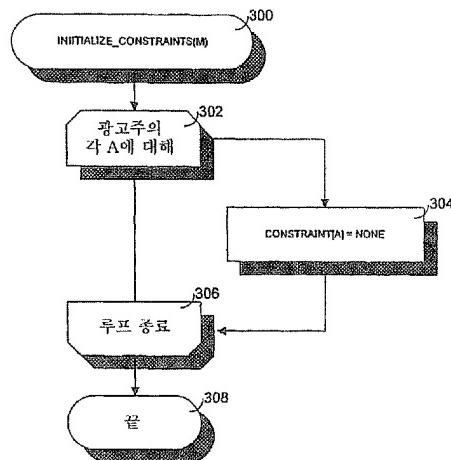
도면 1



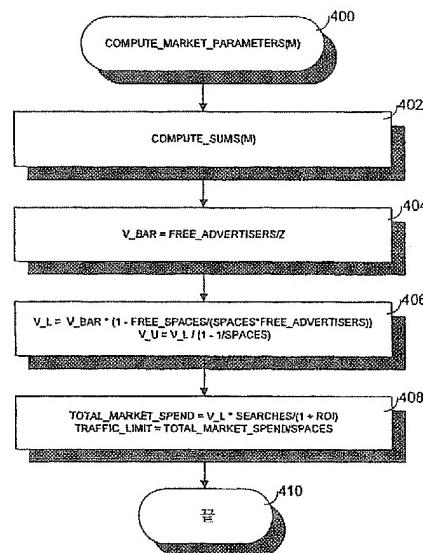
도면 2



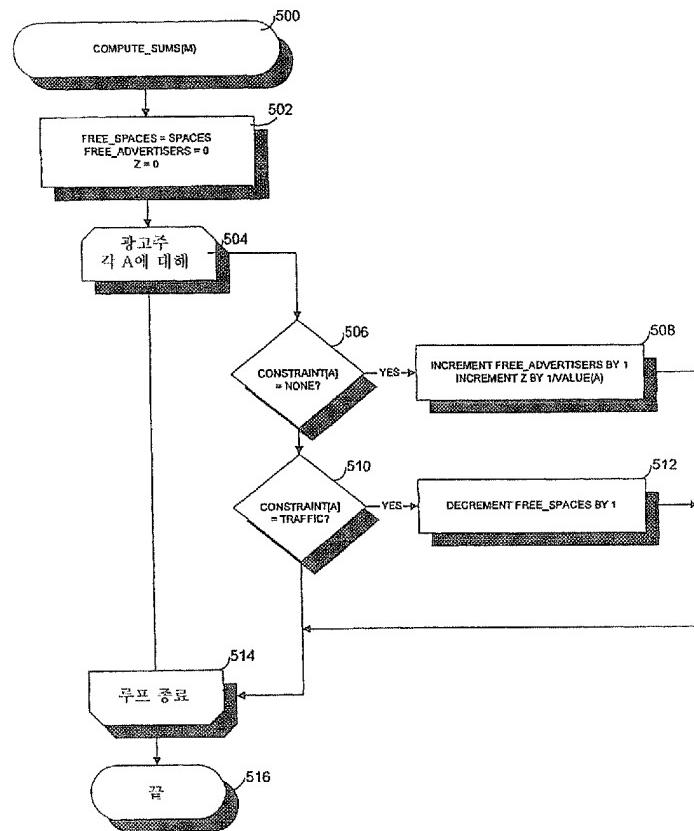
도면 3



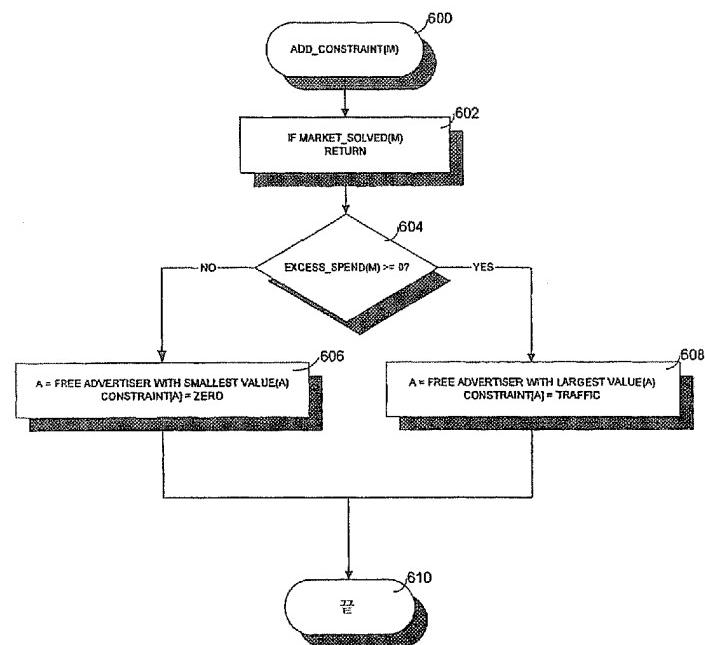
도면 4



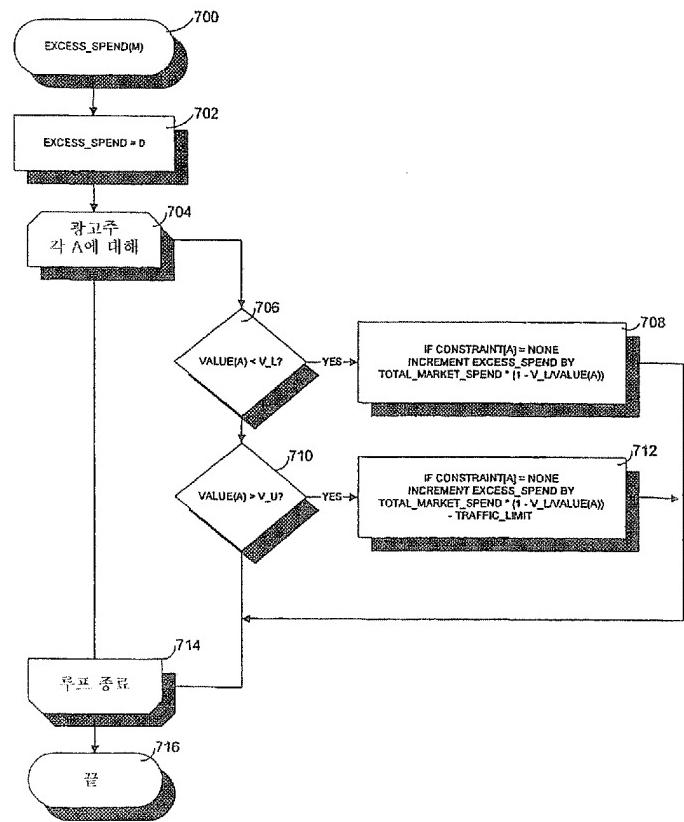
도면5



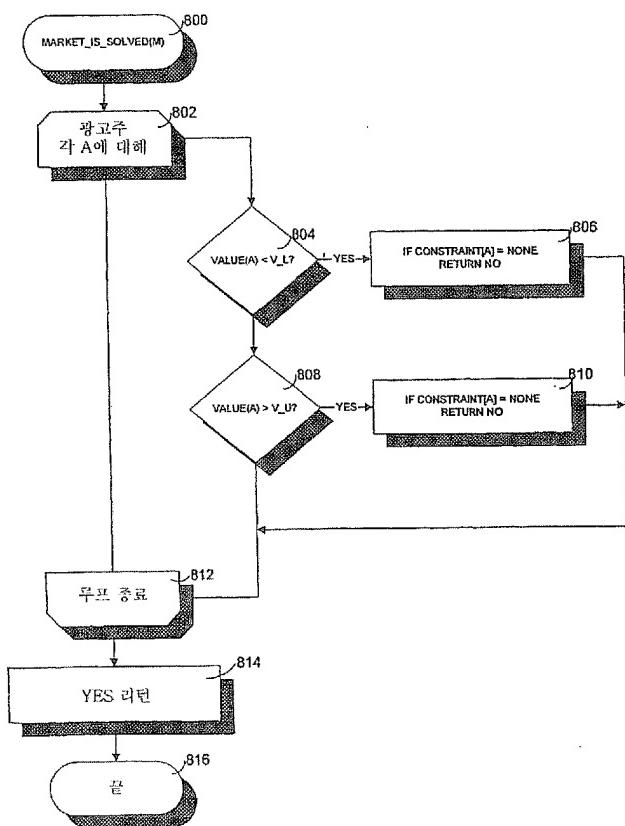
도면6



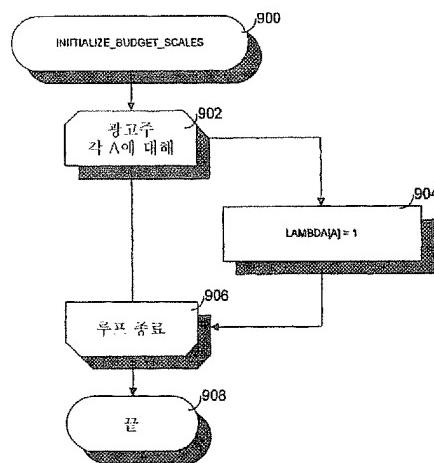
도면 7



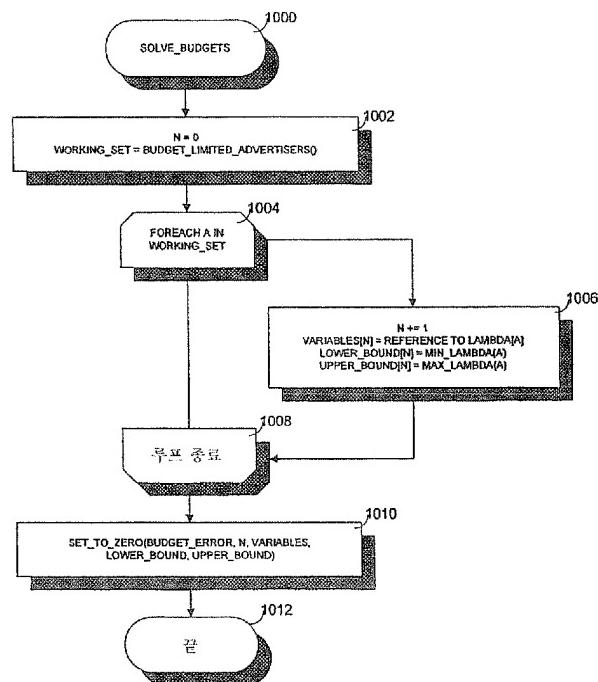
도면 8



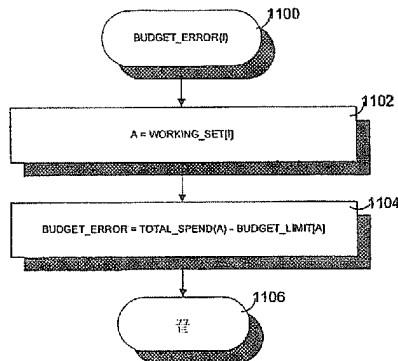
도면 9



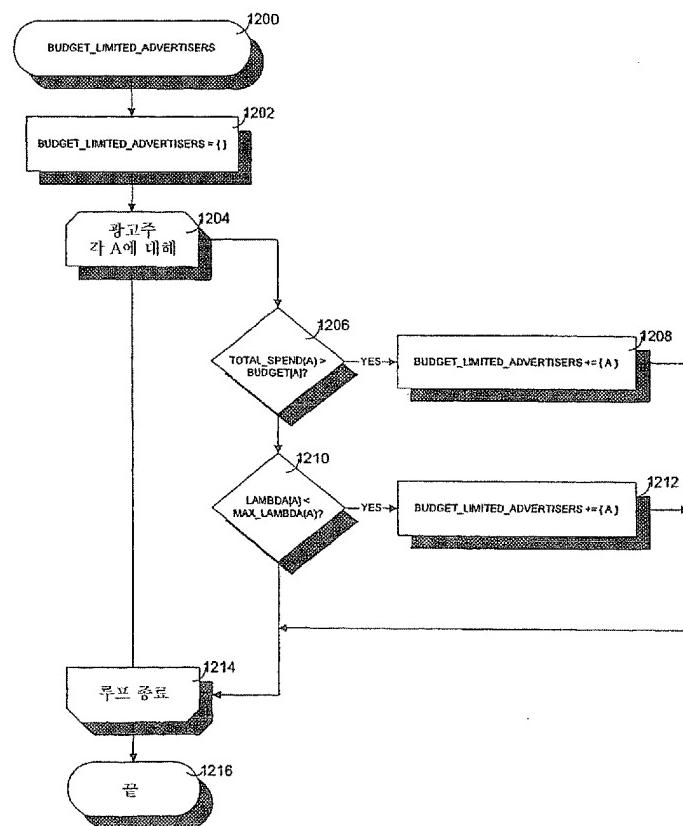
도면 10



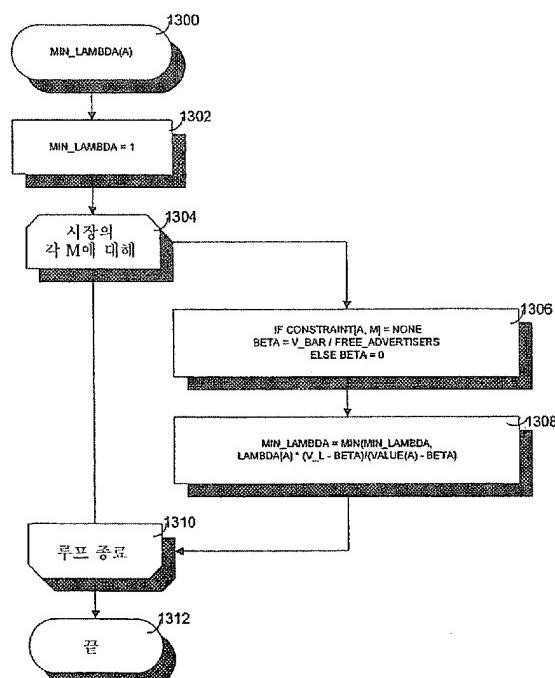
도면 11



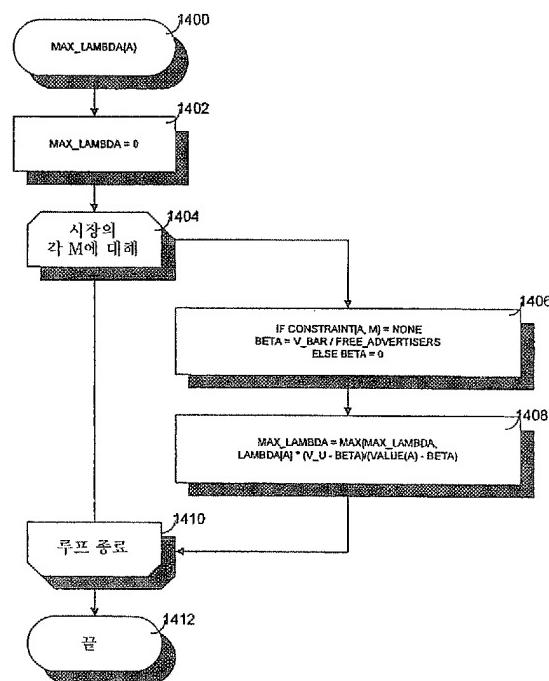
## 5E 면 12



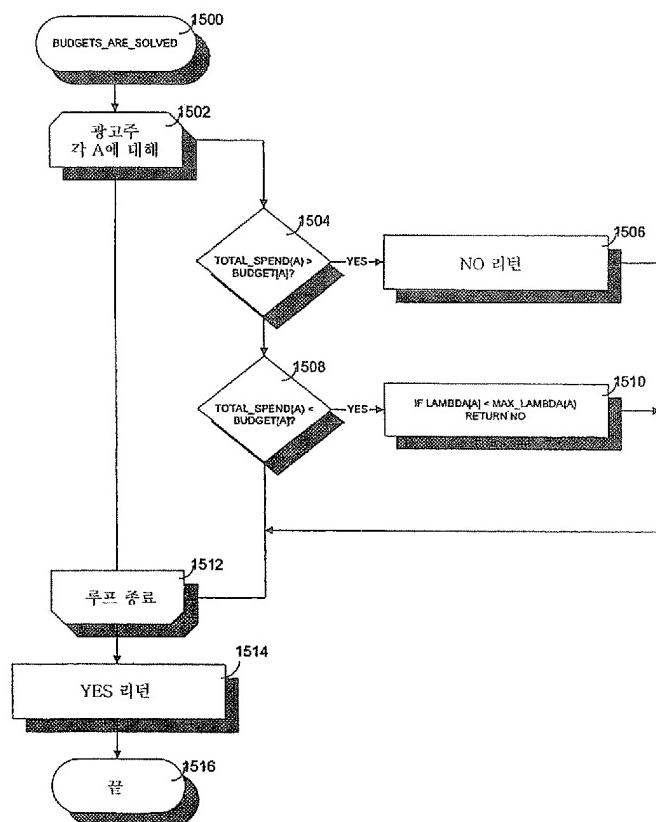
5E 면 13



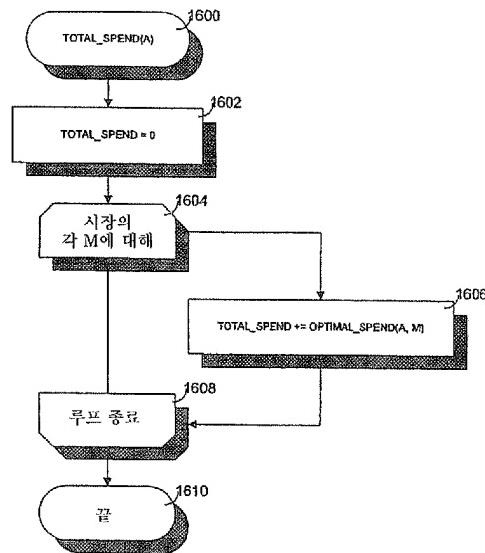
도면 14



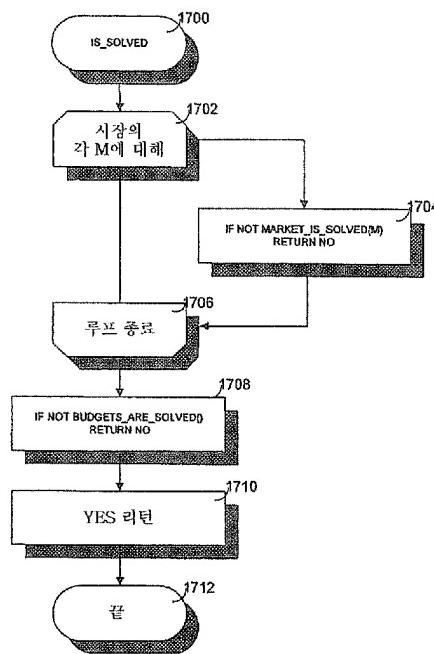
도면 15



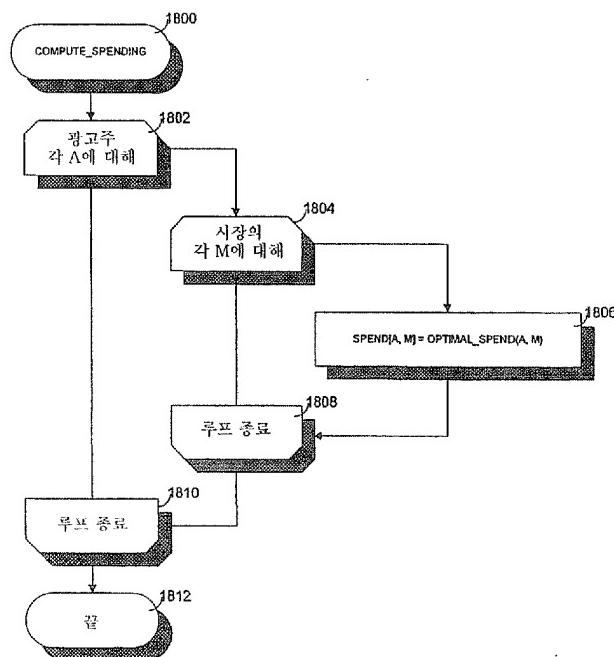
도면 16



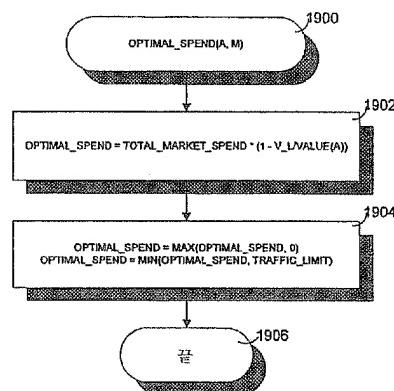
도면 17



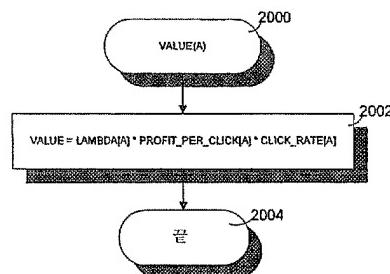
도면18



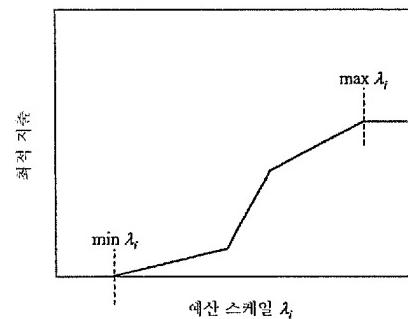
도면19



도면20



도면 21



도면 22

